US006154778A

United States Patent [19]

[11] Patent Number: 6,154,778

[45] Date of Patent: Nov. 28, 2000

Koistinen et al.

[54] **UTILITY-BASED MULTI-CATEGORY QUALITY-OF-SERVICE NEGOTIATION IN DISTRIBUTED SYSTEMS**

[75] Inventors: **Jari Koistinen**, Palo Alto; **Aparna Seetharaman**, Redwood City; **Evan R. Kirshenbaum**, Mountain View, all of Calif.

[73] Assignee: **Hewlett-Packard Company**, Palo Alto, Calif.

[21] Appl. No.: **09/081,265**

[22] Filed: **May 19, 1998**

[51] **Int. Cl.**[7] **G06F 13/00**; G06F 15/16; H04Q 11/00

[52] **U.S. Cl.** **709/228**; 709/227; 709/240; 709/239; 370/230; 370/395

[58] **Field of Search** 709/228, 203, 709/227, 240, 241, 223, 224, 239; 370/395, 409, 230, 465

[56] **References Cited**

U.S. PATENT DOCUMENTS

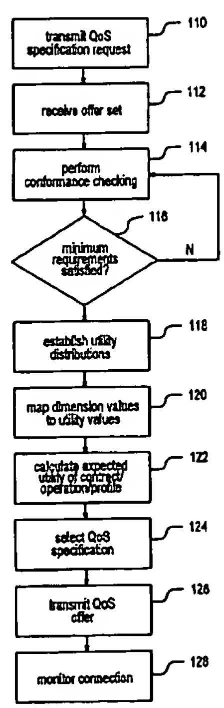
5,065,393	11/1991	Sibbitt et al.	70/58.2
5,408,465	4/1995	Gusella et al.	370/17
5,491,797	2/1996	Thompson et al.	709/200
5,644,715	7/1997	Baugher	395/200.04
5,674,003	10/1997	Andersen et al.	364/514 R
5,732,078	3/1998	Arango	370/355
5,802,058	8/1999	Harris et al.	370/410
5,832,197	11/1998	Houji	714/4
5,892,754	4/1999	Kompella et al.	370/236
5,898,668	4/1999	Shaffer	370/230
5,946,311	8/1999	Alexander, Jr. et al.	370/395
5,948,069	9/1999	Kitai et al.	709/240
5,995,490	11/1999	Shaffer et al.	370/260

Primary Examiner—Glenton B. Burgess
Assistant Examiner—Abdullahi E. Salad

[57] **ABSTRACT**

In a distributed system, a method and system for negotiating a multi-category Quality-of-Service (QoS) agreement between a client and a server includes a client agent enabled to calculate an expected utility to a client of multiple multi-category QoS specifications. The client agent obtains the QoS specifications by transmitting a QoS specification request to a server agent or a broker. The expected utility calculation, based on a probabilistic estimate of QoS levels included in the QoS specifications, enables the client agent to distinguish the QoS specifications of greater value from those of lesser value. The client agent selects at least one of the QoS specifications to be included into an offer for a QoS agreement based on the expected utility calculation. In a preferred embodiment, the client agent selects the QoS specifications determined to be most valuable to the client. The offer is transmitted to the server agent to request a service provided by a server at QoS levels represented by the selected QoS specifications. After transmitting the offer, the client monitors a connection to the server agent for either an acceptance, a rejection, or a counteroffer to the offer. Communication between the client agent and the server agent conforms to a negotiation protocol which provides a set of rules for transmission of negotiation messages.

19 Claims, 12 Drawing Sheets



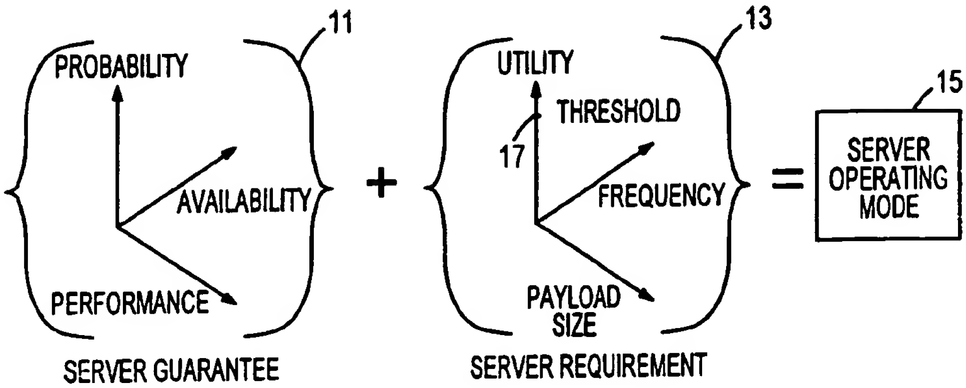

11
PROBABILITY
AVAILABILITY
PERFORMANCE
SERVER GUARANTEE
+
13
UTILITY
THRESHOLD
17
FREQUENCY
PAYLOAD SIZE
SERVER REQUIREMENT
=
15
SERVER OPERATING MODE

FIG. 1

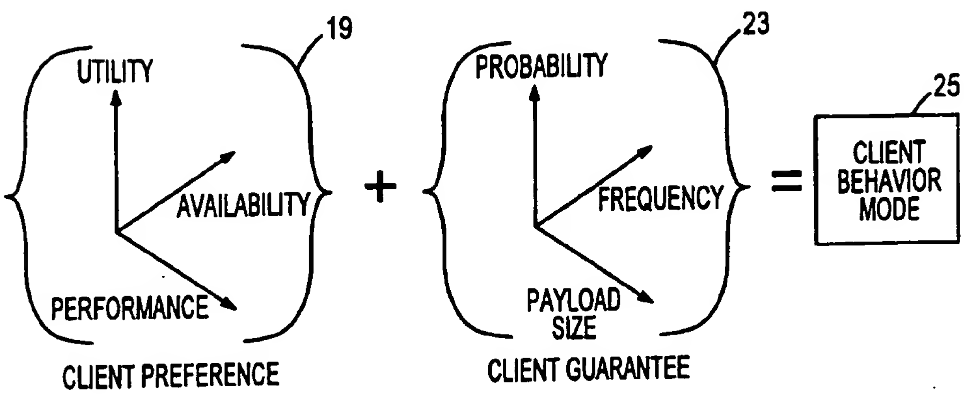

19
UTILITY
AVAILABILITY
PERFORMANCE
CLIENT PREFERENCE
+
23
PROBABILITY
FREQUENCY
PAYLOAD SIZE
CLIENT GUARANTEE
=
25
CLIENT BEHAVIOR MODE

FIG. 2

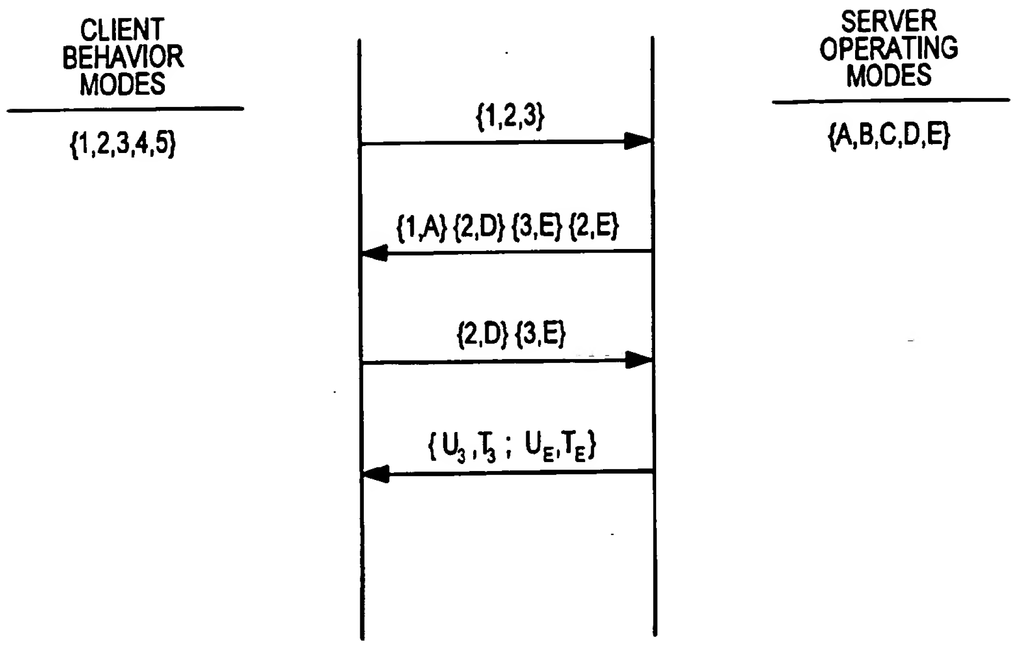

CLIENT BEHAVIOR MODES
{1,2,3,4,5}
SERVER OPERATING MODES
{A,B,C,D,E}
{1,2,3}
{1,A} {2,D} {3,E} {2,E}
{2,D} {3,E}
{ U_3 , T_3 ; U_E, T_E }

FIG. 3

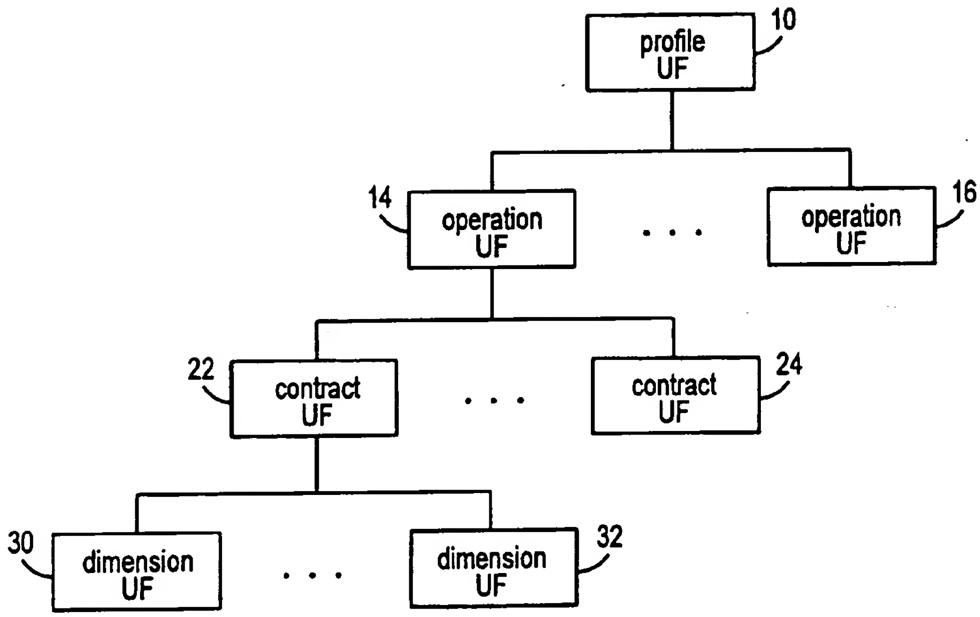

profile UF
10
operation UF
14
. . .
operation UF
16
contract UF
22
. . .
contract UF
24
dimension UF
30
. . .
dimension UF
32

FIG. 4

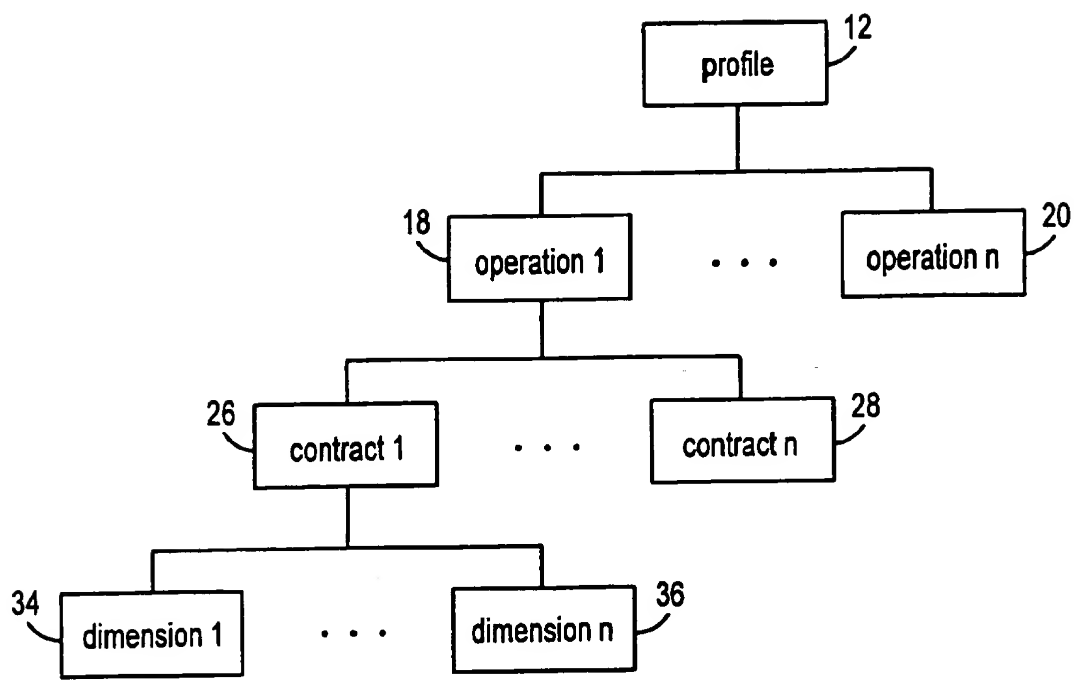

profile
12
18
operation 1
. . .
operation n
20
26
contract 1
. . .
contract n
28
34
dimension 1
. . .
dimension n
36

FIG. 5

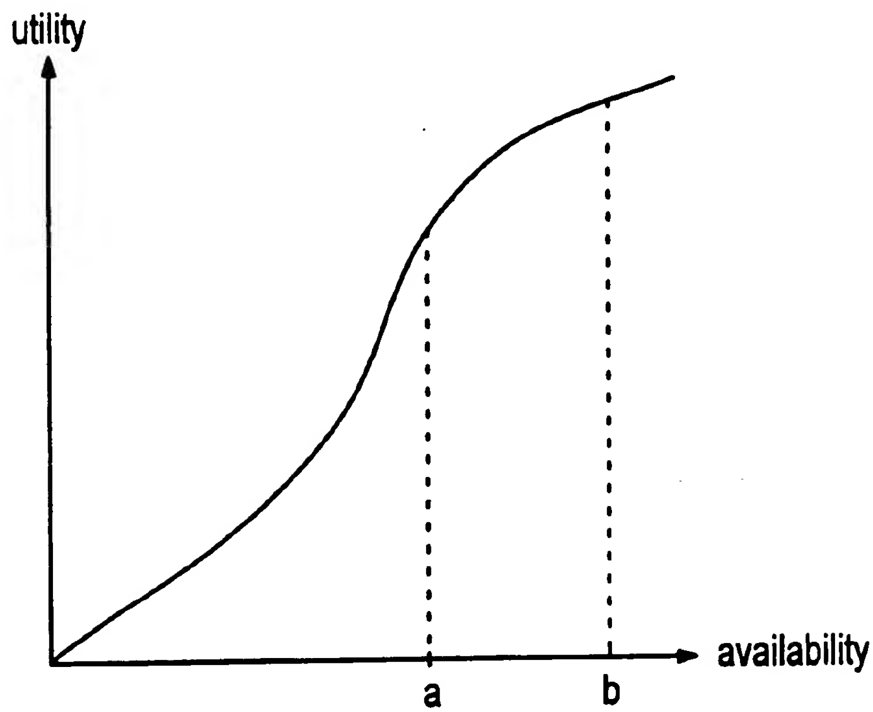


FIG. 6

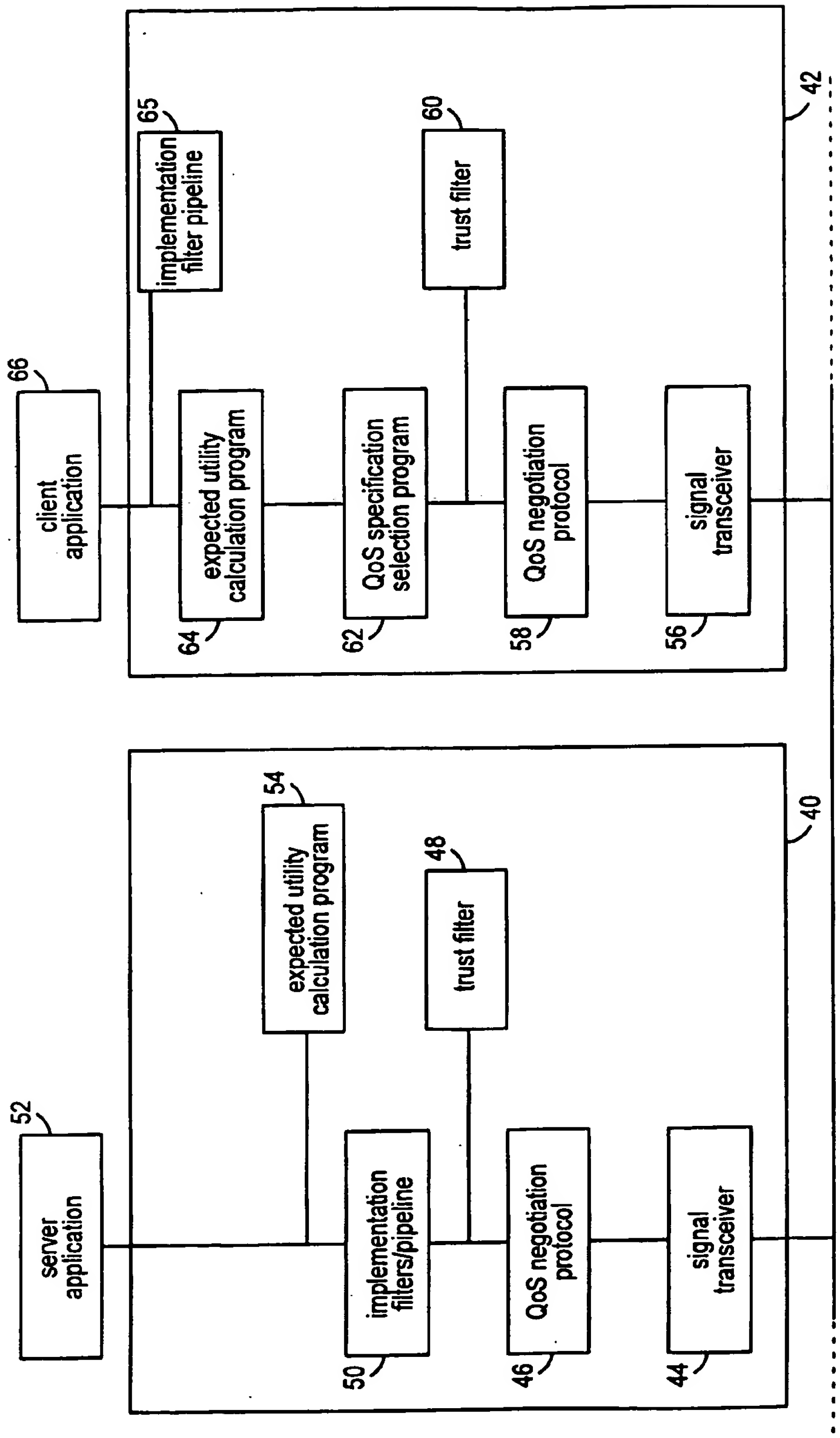

66
client application
65
implementation filter pipeline
64
expected utility calculation program
62
QoS specification selection program
60
trust filter
58
QoS negotiation protocol
56
signal transceiver
42
52
server application
54
expected utility calculation program
50
implementation filters/pipeline
48
trust filter
46
QoS negotiation protocol
44
signal transceiver
40

FIG. 7

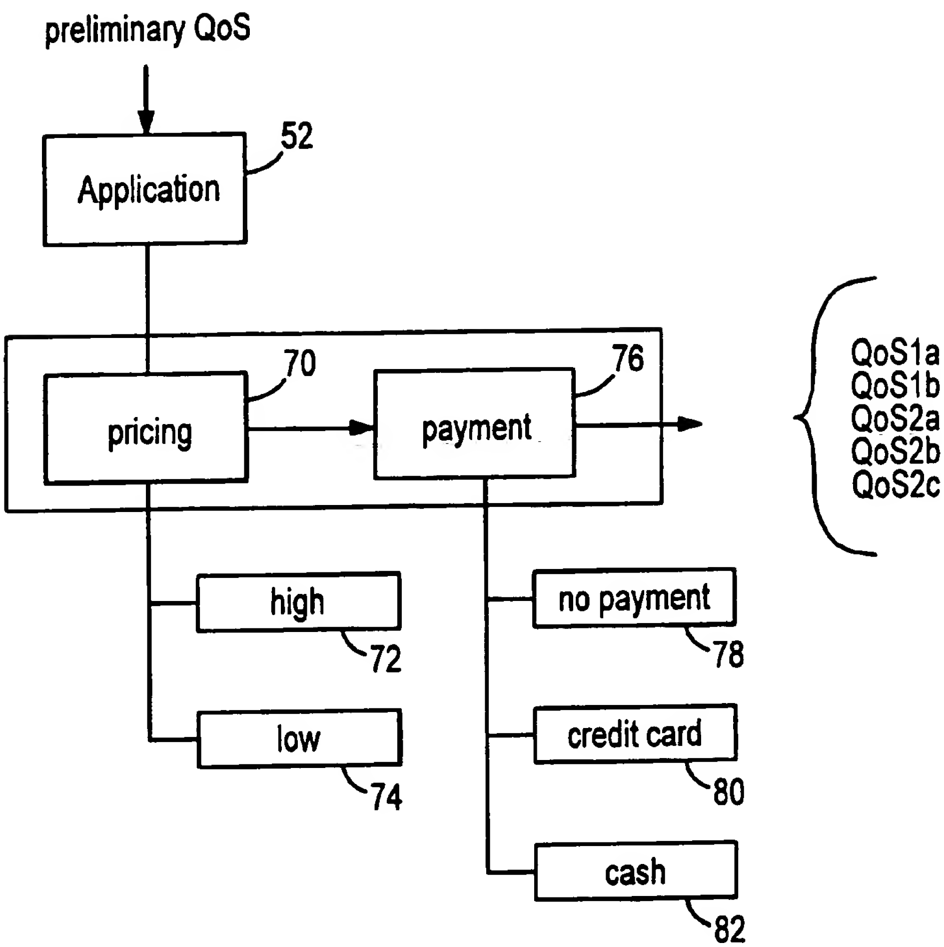

preliminary QoS
Application
52
pricing
70
payment
76
QoS1a
QoS1b
QoS2a
QoS2b
QoS2c
high
72
low
74
no payment
78
credit card
80
cash
82

FIG. 8

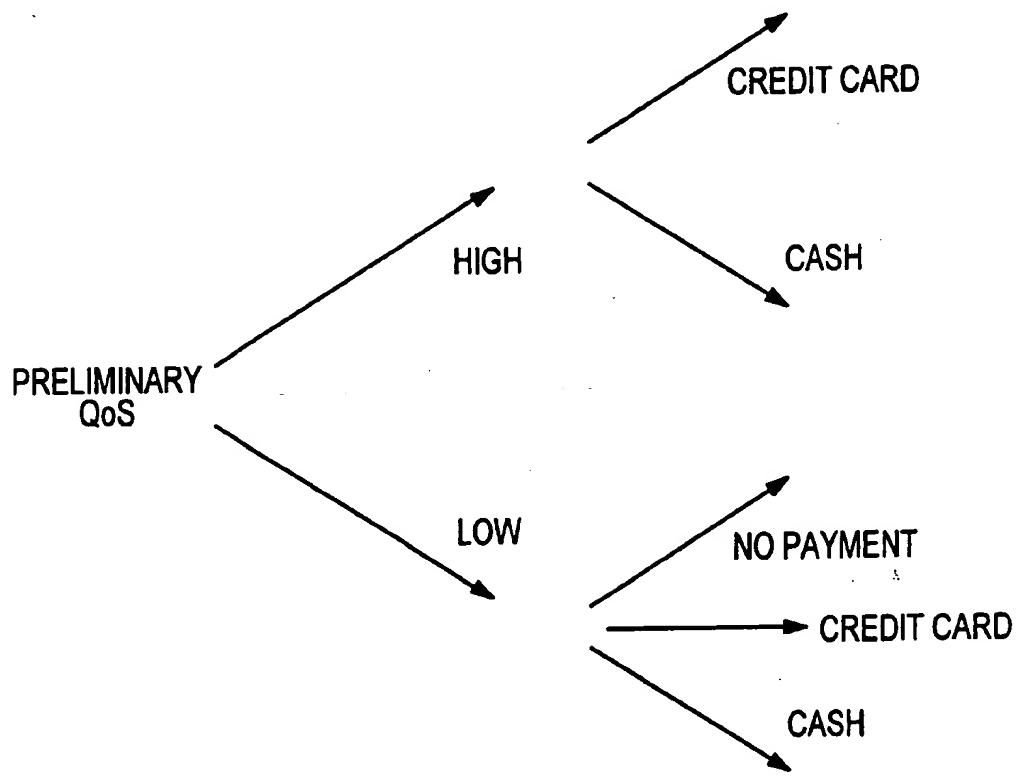

PRELIMINARY QoS
HIGH
LOW
CREDIT CARD
CASH
NO PAYMENT
CREDIT CARD
CASH

FIG. 9

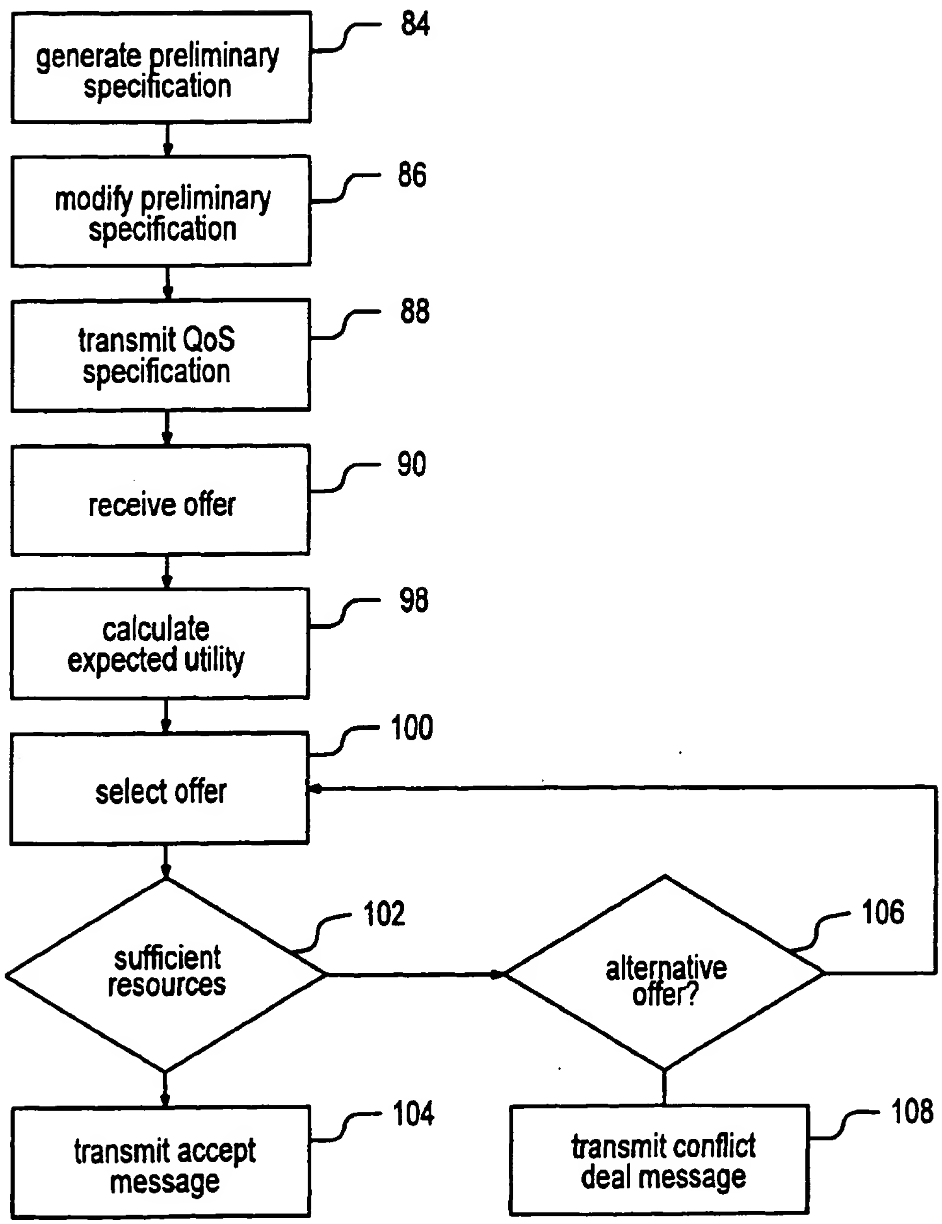

generate preliminary specification
84
modify preliminary specification
86
transmit QoS specification
88
receive offer
90
calculate expected utility
98
select offer
100
sufficient resources
102
alternative offer?
106
transmit accept message
104
transmit conflict deal message
108

FIG. 10

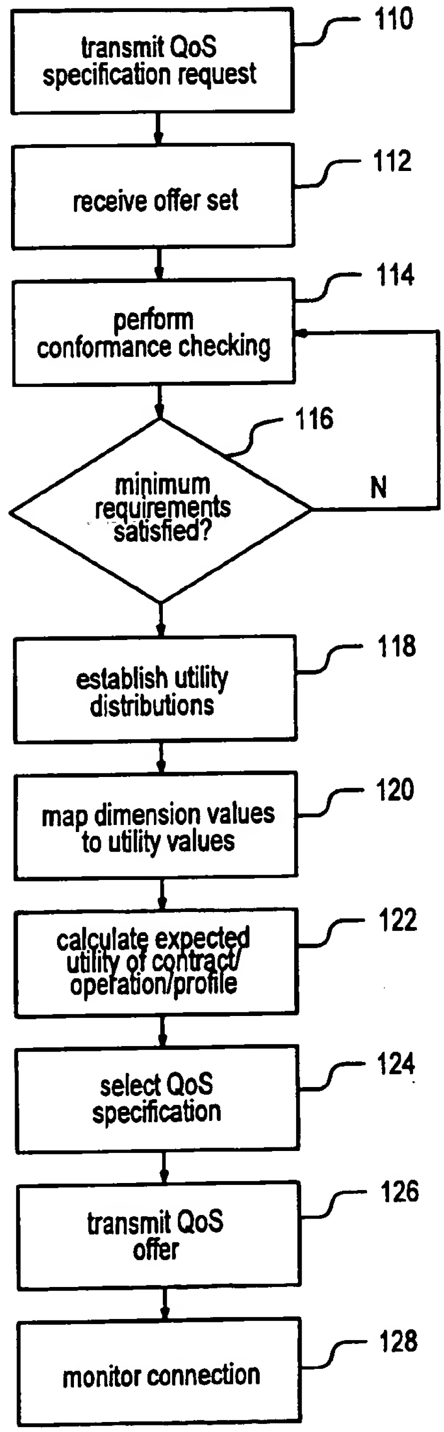

transmit QoS specification request
110
receive offer set
112
perform conformance checking
114
minimum requirements satisfied?
116
N
establish utility distributions
118
map dimension values to utility values
120
calculate expected utility of contract/ operation/profile
122
select QoS specification
124
transmit QoS offer
126
monitor connection
128

FIG. 11

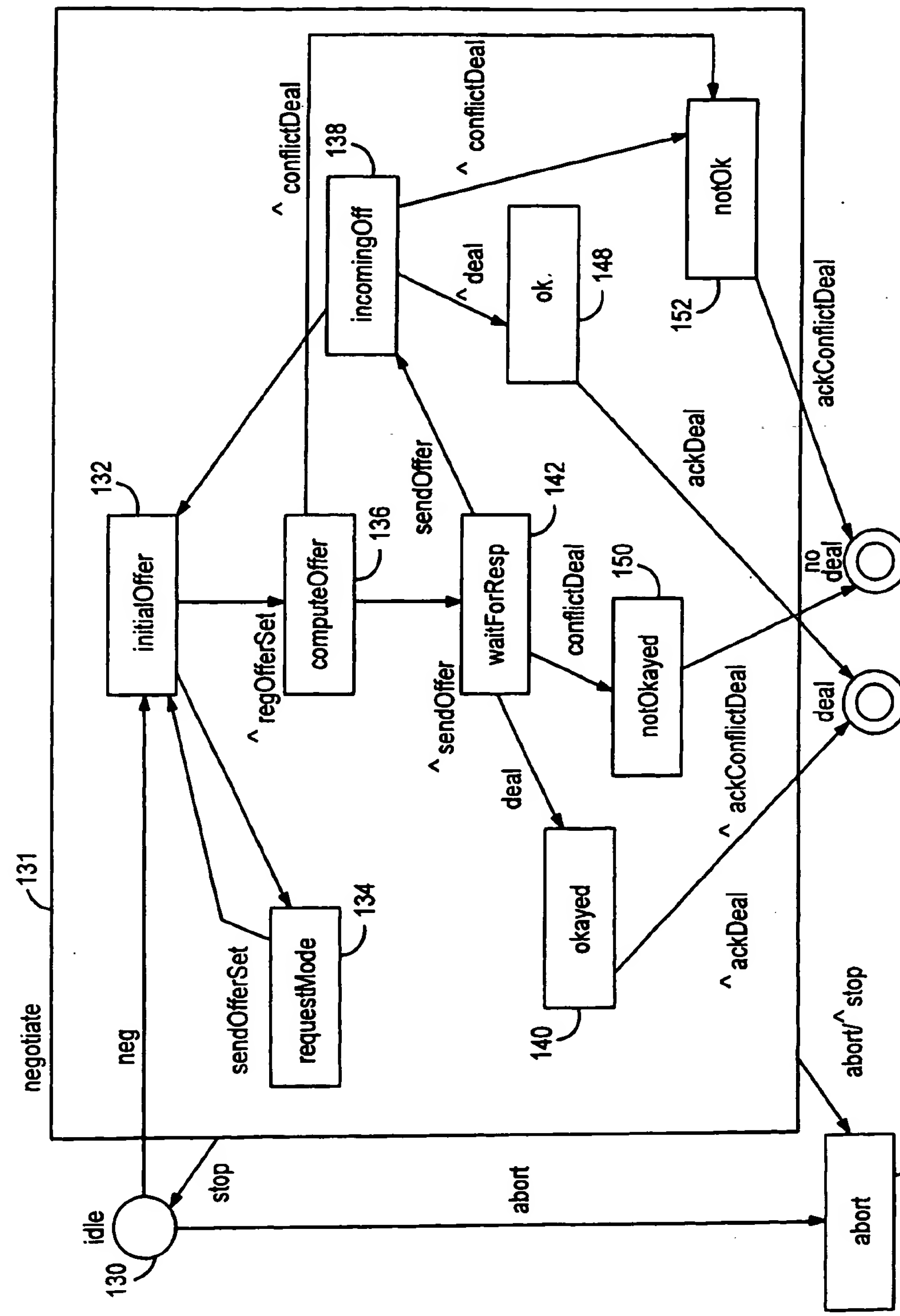

negotiate
131
idle
130
neg
stop
abort
154
abort/^stop
initialOffer
132
requestMode
134
sendOfferSet
^regOfferSet
computeOffer
136
^conflictDeal
waitForResp
142
^sendOffer
sendOffer
incomingOff
138
^conflictDeal
^deal
ok.
148
notOk
152
conflictDeal
notOkayed
150
deal
okayed
140
^ackDeal
^ackConflictDeal
ackDeal
ackConflictDeal
no deal
158
deal
156

FIG. 12

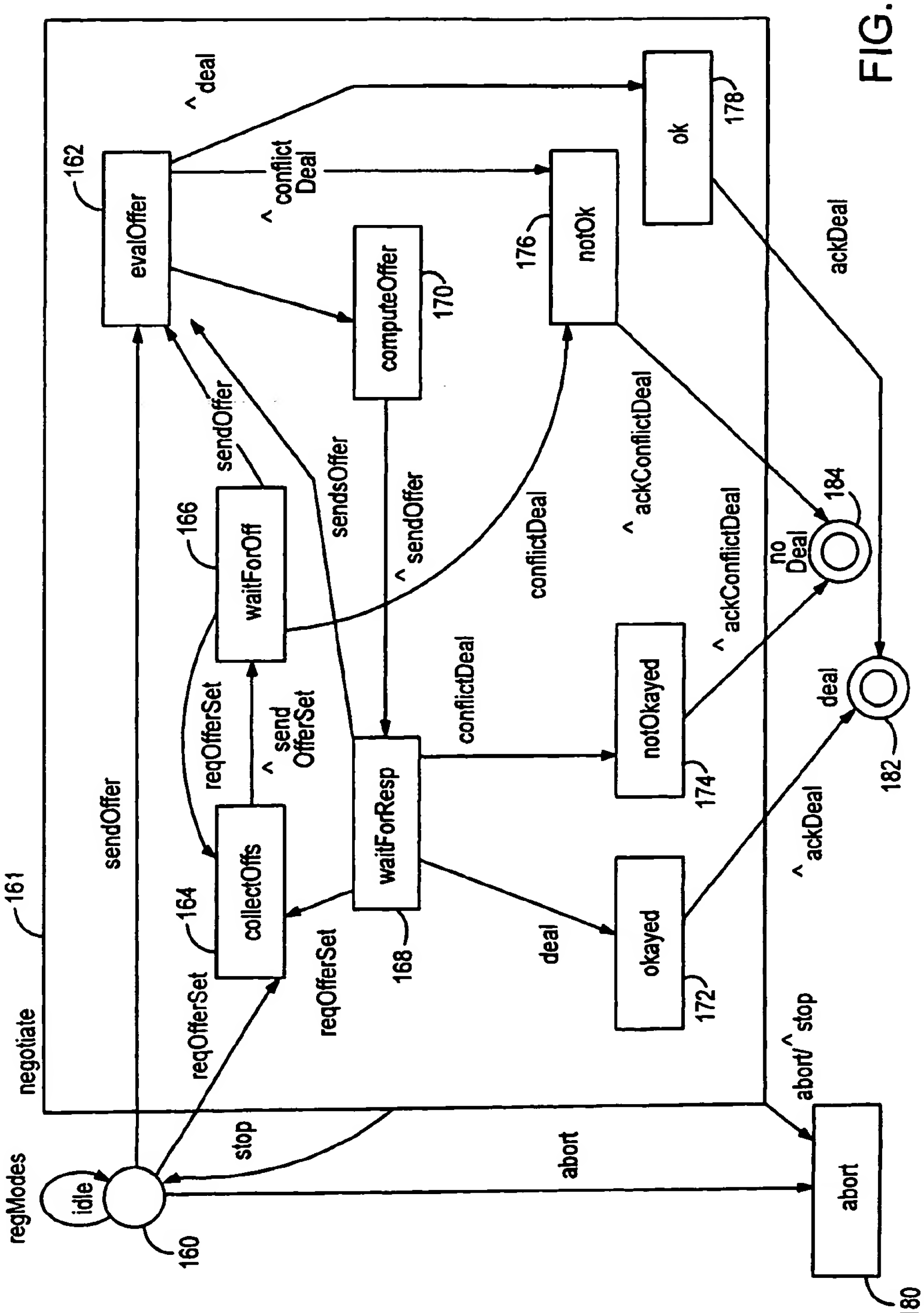

negotiate
161
regModes
idle
160
sendOffer
reqOfferSet
stop
abort
162
evalOffer
^deal
^conflict Deal
ok
178
notOk
176
computeOffer
170
sendOffer
sendsOffer
^sendOffer
conflictDeal
^ackConflictDeal
ackDeal
184
no Deal
166
waitForOff
reqOfferSet
^send OfferSet
collectOffs
164
waitForResp
168
reqOfferSet
conflictDeal
notOkayed
174
^ackConflictDeal
deal
182
deal
okayed
172
^ackDeal
abort/^stop
abort
180

FIG. 13

UTILITY-BASED MULTI-CATEGORY QUALITY-OF-SERVICE NEGOTIATION IN DISTRIBUTED SYSTEMS

TECHNICAL FIELD

The present invention relates generally to a system and method for negotiating a quality-of-service agreement between a client and a server, and more particularly, the invention relates to systems and methods for enabling utility-based multi-category quality-of-service negotiation in distributed systems.

DESCRIPTION OF THE RELATED ART

The term Quality-of-Service (QoS) can be used to describe various non-functional characteristics associated with the operation of communication networks. Enterprises are increasingly dependent on the effectiveness of distributed systems in the daily operation of their businesses. The effectiveness of these distributed systems is in large part dependent on efficient allocation of limited available resources, which can be accomplished through negotiation of QoS agreements. These distributed systems are frequently employed in open network environments, such as the Internet, that have constantly varying levels of QoS. Increasingly mobile workers perform their work from many different connection points and expect applications to handle the varying levels of QoS which are required under different circumstances.

A user of a foreign exchange trading system might access the trading system using a laptop computer at multiple connection points which include a remote dial-up connection and a connection on a local area network (LAN) supporting a server which provides the trading services. The QoS requirements of the user and the capability of the connection to support particular QoS levels can change depending on the connection point of the user. For example, due to geographic proximity, the user's requirement for security will be relatively low when connected directly to the LAN supporting the server as compared to the need for security when connected via the dial-up connection. The dial-up connection might not be able to support as high a performance level as the direct connection on the LAN. Furthermore, resource availability at the same connection point can fluctuate, thereby impacting the user's preferred or required QoS levels with regard to performance and security.

Most existing QoS negotiation protocols are focused on multi-media applications and deal only with a restricted number of QoS parameters. Furthermore, most of these protocols regard negotiation as a process of reserving resources. U.S. Pat. No. 5,674,003 to Andersen et al. describes a computer video conferencing system which employs a socket based transport interface between remote computers over a connection oriented telephony network. A number of sockets are formed into a group and a QoS is associated with the socket group. If a new socket group is established which significantly affects the QoS requirements of the original socket group, a new QoS agreement can be negotiated. The negotiation consists of comparing a QoS requested for use by the new socket group to the QoS available for a telephony connection to the desired endpoint. If the available QoS matches the requested QoS, then the telephony connection is completed. If the available and requested QoS do not match, then a determination is made whether the available QoS is nevertheless acceptable to the socket group. If the available QoS is acceptable, the connection is completed.

Although the Andersen et al. system and method provide a solution to the problem of additional QoS requirements incurred during a call, the system and method is limited to a threshold based negotiation method in which a QoS offer must meet the minimum requirements of a user device in order to be accepted. The method does not enable intelligent selection of a QoS offer from multiple acceptable offers.

U.S. Pat. No. 5,644,715 to Baugher describes a multimedia computer system for scheduling and coordinating multimedia resources. Data structures store user inputs defining scheduling information necessary to support sessions with a specific QoS. In a negotiation for a QoS for a particular session, QoS parameters are specified by a transport user when a connection is requested. If the transport layer immediately recognizes that the values specified by the QoS are not achievable, the communication attempt fails. Alternatively, the transport layer can recognize that it cannot achieve the desired QoS, but it can achieve a lower satisfactory QoS. The lower QoS, minimum acceptable QoS, and maximum acceptable QoS are then transmitted to the transport user as part of a QoS offer. If the user cannot accept the proposed lower QoS, but can accept a value above the minimum or below the maximum, the user may propose an agreement based on some intermediate QoS level. Although the Baugher reference describes a negotiation method for an acceptable QoS when the desired QoS cannot be achieved, the reference does not describe a method by which a server can transmit multiple offers to a client in addition to the maximum and the minimum QoS value offers. The described method does not provide for intelligent selection by the client from among multiple acceptable QoS options.

Furthermore, both the Andersen et al. and the Baugher patents describe a multimedia centric QoS negotiation method which is not compatible with requirements of non-multimedia QoS negotiation.

What is required is a method and a system for efficiently meeting QoS requirements in open networks wherein resource allocation and load vary substantially.

SUMMARY OF THE INVENTION

In a distributed system, a method and system for negotiating a multi-category client-server QoS agreement includes a client enabled to calculate the expected utility of each of a number of multi-category QoS specifications. The QoS specifications are representative of a probabilistic estimate of QoS levels available via a server. The client obtains the QoS specifications in response to transmitting a QoS specification request to the server or a broker. The expected utility calculations enable the client to distinguish the QoS specifications of greater value from those of lesser value. The client selects at least one of the multi-category QoS specifications to be included into an offer for the QoS agreement, based on the expected utility calculations. In a preferred embodiment, the client selects the QoS specifications determined to have the highest expected utility. The offer is transmitted to the server to request a service provided by the server at QoS levels represented by the selected QoS specification(s). After transmitting the offer, the client monitors the connection to the server for either an acceptance, a rejection, or a counteroffer to the offer. Communication between the client and the server during the QoS negotiation conforms to a negotiation protocol which provides a set of rules for the transmission of negotiation messages.

In a preferred embodiment, a set of utility functions is utilized to calculate the expected utility of each QoS specification. The categories of each multi-category QoS speci-

fication include dimensions which represent qualitative or quantitative attributes of a category. For instance, a category of reliability includes the dimensions of availability and mean-time-to-failure. Each dimension of each category in a QoS specification has associated with it a dimension utility function which calculates an expected utility of the dimension. The expected utility for individual dimensions are then combined according to a category utility function to obtain an expected utility for a category.

A preferred embodiment includes a determination of whether the expected utility of each QoS specification satisfies a preselected minimum expected utility threshold. If a particular QoS specification expected utility value does not satisfy the minimum threshold, that specification is not considered for inclusion into an offer for a QoS agreement.

Conformance checking is used for determining whether a particular QoS specification will be considered for inclusion into a QoS offer. Conformance checking involves referencing a constraint profile which defines constraints or requirements which must be met if a QoS agreement is to be reached. If the particular QoS specification does not satisfy the requirements, a utility calculation is not performed on that QoS specification.

To enable application-system transparency, applications associated with the server are enabled to generate preliminary QoS specifications. The preliminary QoS specifications do not reflect the capability of resources to support particular QoS levels because requiring the server applications to possess such detailed low level information would create a substantial burden on the applications. The application-generated preliminary QoS specifications are then filtered to modify the dimension values according to capabilities of resources of the distributed system to support QoS levels. Filtering a preliminary QoS specification can generate multiple QoS specifications.

The calculation of expected utility can include filtering the data included in the QoS specification utilizing a client trust filter which modifies data in the QoS specification according to a level of trust the client possesses with regard to the server. The trust level can be based on the server's performance under previous QoS agreements with the client. Alternatively, the client can access server trustworthiness data from a reputation broker. The broker can be configured to assemble server trustworthiness data by either collecting the data from clients or by entering into QoS agreements with the server to determine how closely the server's actual behavior corresponds to its promised behavior.

The server can be equipped with a server trust filter to process offers from the client to determine whether to accept the offers. Each QoS specification, in addition to containing a server promise to provide a service at certain QoS levels, also includes a corresponding client promise to conform client behavior to certain server requirements. The server trust filter modifies the promise by the client to behave according to the server requirements to reflect the behavior which the server actually expects. Accordingly, the server might reject an offer which, if the stated promises for client behavior were believed, the server would accept. The server can access trust data regarding the client based on the extent to which the terms of previous QoS agreements negotiated by the client have been satisfactorily executed. The data can be collected by the server or it can be accessed from a reputation broker.

The client utilizes an expected default provision function in the calculation of the expected utility for dimensions which are not referenced in QoS specifications. The expected default provision function provides a probabilistic estimate of server QoS behavior with regard to a dimension absent from a QoS specification. The server utilizes an expected default interaction function to provide a probabilistic estimate of client behavior for an aspect of client behavior of interest to the server which is absent from a client behavior mode. The expected default provision functions can be based on past QoS behavior of the server with regard to particular dimensions, the identity of the server, and the server's guarantees. Likewise, the expected defaullt interaction function can be based on past client behavior, client identity and/or client preferences.

The server preferably employs a handler class to determine whether system components are capable of supporting QoS levels specified for those dimensions based on commitments previously assigned to the components. The server invokes the handler class to determine whether to accept a particular offer. Even though the server has filtered the preliminary application-generated QoS specification to generally reflect the capabilities of the system components to support specific QoS levels, the previously assigned commitments, which are not taken into account in the filtering process, might prevent the components from supporting the modified QoS levels.

The present invention provides the advantage of an application-independent generic negotiation protocol. That is, the negotiation protocol is not limited to multi-media applications or any other type of application. Another advantage lies in the efficient allocation of system resources resulting from a QoS agreement narrowly tailored to server capabilities and client requirements. Yet another advantage is that applications are enabled to offer a service in multiple QoS modes. Another advantage is that the invention provides application-system transparency by allowing applications to generate QoS specifications independently of the environment and system in which they operate.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic representation of a server guarantee and a server requirement which comprise a server operating mode in a utility-based embodiment of the present invention.

FIG. 2 is a schematic representation of a client preference and a client guarantee which comprise a client behavior mode in the utility-based embodiment.

FIG. 3 illustrates an exchange of server operating modes and client behavior profiles shown in FIGS. 1 and 2 during a QoS negotiation.

FIG. 4 is a schematic representation of a utility function profile for calculating an expected utility of a QoS specification in a utility-based embodiment of the QoS negotiation.

FIG. 5 is a schematic representation of a QoS specification in a utility-based embodiment for a QoS negotiation.

FIG. 6 is a graph of a utility distribution for an availability dimension of the QoS specification shown in FIG. 5.

FIG. 7 is a block diagram of a server agent and a client agent according to the present invention.

FIG. 8 is a block diagram of an embodiment of a filter pipeline illustrating the cooperation of filters in processing a preliminary QoS specification.

FIG. 9 is a schematic representation of a disjunctive representation of a filtered QoS specification.

FIG. 10 is a process flow for an operation of the server agent in FIG. 7.

FIG. 11 is a process flow for an operation of the client agent in FIG. 7.

FIG. 12 is a state diagram illustrating the various operational states of the client agent during a negotiation.

FIG. 13 is a state diagram illustrating the various operational states of the server agent during a negotiation.

DETAILED DESCRIPTION

Referring to FIG. 1, a utility-based embodiment of the multi-category QoS negotiation includes a server operating mode 15 which includes a server guarantee 11 and a server requirement 13, which essentially is a utility function. The server guarantee 11 expresses QoS levels in the form of a probability distribution. In FIG. 1 the server guarantee 11 is represented as a graph, correlating dimension values for server availability and server performance to probability values. Although two dimensions are described in FIG. 1, the number of dimensions included in a server guarantee can vary depending on the capabilities and requirements of the server. The server guarantee 11 is a probabilistic estimate of QoS levels at which the server can provide a particular service. The probabilistic estimate of the server's QoS behavior in the guarantee 11 is not necessarily expressable as a linear combination of the probabilistic estimates of multiple dimensions if the guarantee references multiple dimensions. For instance, the combination availability of greater than 95% and a performance of 200 Kbps might correspond to a probability of 87%. In other words, there is an 87% likelihood that the server will be available for more than 95% of calls made by a client when providing a service at a rate of 200 Kbps. However, the server guarantee 11 is conditional on client behavior conforming to certain requirements represented in the server requirement 13.

The server requirement 13 correlates client behavior to utility values for the server and can be combined with a probability distribution of a client behavior mode to calculate an expected utility of a client operating mode. Because client behavior in a client operating mode is expressed in probabilistic terms, the server requirement produces an expected utility of the client operating mode which is expressed in probabilistic terms. In FIG. 1 the requirement function 13 correlates frequency of client calls and payload size of packets transmitted by the client to expected utility to the server. Although two aspects of client behavior are included in the server requirement in FIG. 1, more or fewer aspects may be included. The server requirement 13 may include a threshold 17 which must be satisfied if the server's estimated QoS behavior is to be considered a guarantee. Specifically, the combination of client call frequency and the payload size must satisfy the expected utility threshold The server requirement 13 is not necessarily monotonic with regard to a particular client behavior attribute. For instance, within a range of call frequency between one call per hour and twenty calls per hour, the expected utility does not necessarily only increase or only decrease. The expected utility might decrease between one and seven calls per hour, increase between seven and twelve calls per hour, and decrease again between thirteen and twenty calls per hour.

With reference to FIG. 2, a client behavior mode 25 includes a client preference 19 and a client guarantee 23. The client preference 19 correlates dimension values of a service provided by a server to an utility values to the client. The client preference 19 is not necessarily monotonic with regard to any particular dimension. Furthermore, the client preference 19 might calculate expected utility in a non-independent manner with respect to the dimensions of the server guarantee 11. The client preference 19 can be combined with the server guarantee to provide an expected utility to the client for the server guarantee.

The client behavior mode 25 further includes a client guarantee 23 which expresses client behavior in terms of a probability distribution. In FIG. 2, the client guarantee correlates client call frequency and payload size to probability values. Although the client guarantee 23 in FIG. 2 includes two client behaviors, the client guarantee can include fewer or more than two behaviors. The server requirement 13 can be utilized to determine an expected utility to the server of the client guarantee 23.

The server operating mode 15 might be one of multiple server operating modes which represent different QoS levels at which a server can provide a particular service. The server operating modes collectively can be viewed as an invitation to clients to negotiate a QoS agreement. Essentially, the server operating modes communicate to clients that the server is willing to provide a service at particular QoS levels according to probability distributions on the condition that client behavior satisfies minimum expected utility thresholds as expressed by server requirements. The client behavior mode 25 might be one of multiple client behavior modes which represent client behavior to which the client is willing to conform.

Referring to FIG. 3, a simplified example of a utility-based negotiation for a multi-category QoS agreement between a server and a client includes a set of client behavior modes {1,2,3,4,5} and a set of server operating modes {A,B,C,D,E}. In a preferred embodiment, the client behavior mode set is included in a client agent which negotiates on behalf of a client with a server agent. The server agent includes the server operating mode set and negotiates with the client agent on behalf of the server. In an alternative embodiment, the client and server negotiate the QoS agreement themselves. In a first transmission to a server agent (not shown), a client agent (not shown) transmits a subset {1,2,3} of the set of client behavior modes {1,2,3,4,5} in a request for offers from the server agent. In an alternative embodiment, the client agent transmits the request to a broker which handles the offer on behalf of the server agent. The server might provide a broker with server operating modes which the broker advertises on behalf of the server. The selection of the client behavior modes might be based on a behavior mode preference function which represents the client's behavior preference. The request for offers does not necessarily include behavior modes, but in a preferred embodiment the behavior modes are included so that the server agent can select a subset of operating modes based on the behavior modes to be included in a set of offers {1,A}, {2,D}, {3,E} and {2E}, which are transmitted to the client agent. In FIG. 3, each offer includes a behavior mode and an operating mode. However, a server requirement is not necessarily included in the server operating mode which the server includes in each offer. Furthermore, if the client agent does not transmit the behavior modes in the offer request, the offer set might consist solely of server guarantees. The server agent can determine which operating modes to include in the offers based on the minimum expected utility thresholds of the server operating modes.

If the client agent includes the behavior modes in the request for offers, the behavior modes might include the client preference if it is assumed that the client agent and server agent negotiate in an environment of trust. Trust implies that the stated values in the server guarantee are the values the server intends to abide by, and the stated values in the client guarantee are the values the client intends to abide by. Including the client preference enables the server to select only those operating modes which will satisfy the client's requirements, thereby expediting the negotiation

process. Even in the absence of trust, the behavior modes in the request might include the client preference. However, as will be discussed further below, the server will pass the utility function through a trust function.

If the client agent does not include the client preference 19 in the request for offers, the client agent utilizes the client preferences to calculate an expected utility of the offers and the client agent performs a ranking of the offers. The client agent might utilize a global utility threshold which the client agent utilizes to determine if any offer exceeds a minimum expected utility. The client agent preferably selects a subset of the offers which represent the offers with the highest expected utility to the client. Alternatively, the client agent might simply select the first n offers which exceed the minimum threshold, with n being some positive integer. In FIG. 3, the client agent selects offers {2,D} and {3,E} to transmit to the server agent as offers for a multi-category QoS agreement. Upon receiving the client-selected offers {2,D} and {3,E}, the server agent ranks the offers and selects the offer which is most desirable to the server. The ranking can be based on the expected utility of the client behavior modes included in the offers. The server can also utilize an operating mode preference function which represents the server's preference for choosing a particular operating mode. If the server agent accepts an offer, the server agent transmits an acceptance message $\{U_3,T_3,; U_E,T_E\}$ which indicates acceptance of the offer {3,E}. The agreement is expressed in terms of the client preference (U_3,T_3) and the server requiremtnt (U_E,T_E) to enable verification of satisfactory client and server performance under the contract. Verification of satisfactory performance is performed by determining whether the client's behavior satisfies the server expected utility threshold and whether the QoS provided by the server satisfies the global client expected utility threshold.

In an alternative embodiment, a client formulates an offer for a QoS agreement in the absence of obtaining any server operating modes from the server agent. The client offer includes a server operating mode 15 and a client behavior mode 25 in the client-generated offer. The offer is included in a first call from the client under the proposed QoS agreement. The first call indicates that acceptance of the QoS offer can be provided by the server processing the first call according to the terms of the proposed QoS agreement. In this manner, the negotiation can be streamlined to maximize efficiency.

In a preferred mode, the calculation of expected utility by the server agent includes utilizing an expected default interaction function. This function is utilized by the server agent when the client guarantee 23 is silent regarding a particular client behavior parameter which is of interest to the server agent. The expected default interaction function enables an expected utility calculation for the particular client behavior parameter given the operating mode which the server agent is proposing. The expected default interaction function can reflect the server agent's past interactions with the client agent, or it might reflect past client behavior data which the server agent received from a reputation broker, not shown. The expected default interaction function can also be parameterized on the client's identity and the client's guarantee.

A related function employed by the server agent in the expected utility calculation process is a trust function. The client agent is capable of intentionally misrepresenting what the client's behavior will be under a proposed QoS agreement. The trust function is representative of a level of trust which the server agent possesses with regard to the client agent and it is used to modify values in the client guarantee which results in a modification of the calculated expected utility. For instance, the client agent might represent that the client will make no more than ten calls in any 24-hour period under the proposed agreement. However, the server agent might have information which indicates that the client has broken that same promise in the past 90% of the time. Consequently, even if the server agent would have accepted the client agent's offer if the promise were taken at face value, upon filtering the offer with the trust function, the server agent might reject the offer. Client agent trustworthiness data which the trust function utilizes can be based on past interactions the server agent has had with the client agent, or the data can be obtained from a reputation broker, not shown. The reputation broker has either polled the trustworthiness data from other servers or has entered into QoS agreements itself with the client agent to determine the client agents trustworthiness. A trust function can also be utilized to filter information provided by the reputation broker based on the trustworthiness of the broker.

In a preferred embodiment, the calculation of expected utility by the client agent includes utilizing an expected default provision function if the server guarantee does not include a reference to a dimension of interest to the client. The expected default provision function provides expected server QoS behavior with regard to the dimension of interest in the same form as would be included in a server guarantee 11. The client agent uses this expected QoS behavior information in the expected utility calculation. The expected server QoS behavior can be based on the server's past identity, the server's behavior and the server guaranteed 11.

The client agent also preferably utilizes a trust function in the calculation of expected utility of server operating modes. The client agent's use of the trust function is analogous to that of the server agent. The server agent is also capable of intentionally misrepresenting what the server's QoS behavior will be under a proposed QoS agreement. The trust function is utilized to modify a declared server guarantee to reflect the believed server guarantee. For example, if the mean availability specified in a server guarantee is X, the trust function might reduce the value to a believed value of 0.8X. The trust function can be based upon past interactions with the server agent, the server's identity, and the server's guarantees. Alternatively, the server agent trustworthiness data is assembled by a reputation broker which has either polled the trustworthiness data from other servers or has negotiated QoS agreements with the server agent to determine the server's trustworthiness directly. The broker then transmits the trustworthiness data to the client agent upon request.

Another embodiment of the trust function can include a certification process whereby a third party certifies a level of trustworthiness of agents. Upon initiating a negotiation, the server agent and the client agent produce their respective trustworthiness certifications. The server agent might calibrate its trust function according to the client agent's certification and the client agent might calibrate its trust function according to the server agent's certification.

Referring to FIGS. 4 and 5, schematic representation of a utility-based embodiment of a multi-category QoS negotiation includes a profile utility function 10 associated with a profile of a multi-category QoS specification 12. The multi-category QoS specification is generated by a server (not shown) to represent to a client (not shown) the QoS levels at which the server is able to provide services. As will be described in greater detail below, the client utilizes the profile utility function 10 to calculate the expected utility to the client of the QoS specification. The QoS specification is

arranged in a hierarchical manner, and consequently, a utility profile which defines all of the utility functions utilized to calculate the expected utility also has a hierarchical organization.

The profile utility function 10 calculates the expected utility of the QoS specification by applying weights to the expected utility values of n operations. The operations include a first operation 18 through n^{th} operation 20, with n representing any positive integer. The expected utilities of the operations are calculated by first and second operation utility functions 14 and 16. The first operation 18 might be a buy operation and the n^{th} operation a sell operation of an on-line foreign exchange trading service. If the buy operation is more important to the client than the sell operation, the client profile utility function 10 assigns a higher weight to the buy operation than it assigns to the sell operation.

Each operation includes at least one contract, with each contract representing a category of the QoS specification such as reliability, performance, or security. The buy operation 18 includes a first contract 22 through an n^{th} contract 24, with n representing any positive integer. The buy operation utility function 14 calculates an expected utility for the buy operation by assigning weights to the various contracts 26 and 28 included in the buy operation. In FIG. 1 the first contract represents the QoS category of reliability and the n^{th} contract 28 represents the QoS category of performance. If the client desires performance over reliability, the operation utility function 14 weights the performance contact expected utility more heavily than the reliability contract expected utility.

Although both the profile utility functions and the operation utility functions have been described as calculating expected utility by applying weights to operations and contracts respectively, this is not critical to the invention. The profile and operation utility functions can utilize other more sophisticated methods for calculating expected utility.

The reliability contract 26 includes dimension 1 through dimension n, 34 and 36, with n representing any positive integer. The expected utilities of dimension 1 and dimension n are calculated utilizing a dimension 1 utility function 30 and a dimension n utility function 32. The first dimension 34 may represent a dimension called availability, which is part of the reliability contract 26. Referring briefly to FIG. 2, a graph representative of an embodiment of the availability utility function is shown. Assuming that the QoS specification guarantees the availability will be between a and b with a generally uniform distribution, the expected utility may be the mean of the integral for the interval between a and b.

Referring to FIG. 7, a server agent 40 and a client agent 42 provide QoS negotiation functionality on behalf of a server and a client, respectively. It should be understood that both the server agent 40 and the client agent 42 are not necessarily contained in stand-alone devices separate from the server and the client. It is possible that the functional components of both agents are integrated into software and distributed throughout the network on which they reside. The server and client agents 40 and 42 might alternatively be incorporated into the server and client, respectively.

The server agent 40 is associated with a server application 52. The QoS negotiation centers around the QoS levels at which the server can provide a service for the client and the client behavior modes on which the QoS levels will be conditioned. The server application 52 is configured to generate preliminary QoS specifications for services supported by the server. The preliminary QoS levels represented in the preliminary QoS specifications might not reflect the environment in which the server application 52 operates. That is, the server application 52 might not be configured to factor in the resources of the distributed system when formulating these preliminary QoS specifications. Requiring the application 52 to determine the QoS that its environment can support places a burden on the application and it requires the application to be aware of low-level system details. Consequently, if the server application 52 does not take system resources into account, the preliminary QoS specifications need to be modified to more accurately reflect the QoS levels which the system resources are capable of supporting.

A server implementation filter pipeline 50 enables network components to modify the preliminary QoS specification to accurately reflect the capabilities and requirements of those components. Referring now to FIG. 8, an example of an implementation filter pipeline is shown wherein the application 52 is connected to a series of two filter types, or meta-filters, which include a pricing meta-filter 70 and payment meta-filter 76. Each meta-filter is associated with a distributed system component which supports some aspect of a QoS specification. A particular system component might be associated with multiple filters. For instance, a payment system component (not shown) determines a client payment method for payment of services rendered by the server under the QoS agreement. Associated with the payment meta-filter 76 might be three payment filters; a no payment filter 78, a credit card filter 80, and a cash payment filter 82. A pricing meta-filter includes a high price filter 72 and a low price filter 74.

A preliminary multi-category QoS specification generated by the application 52 includes some price for a service. The pricing meta-filter 70 transmits the preliminary QoS specification to both the high price filter 72 and the low price filter 74 for processing. The preliminary QoS specification is modified to produce filtered specifications QoS1 and QoS2. The filtered specifications QoS1 and QoS2 are then collected by the pricing meta-filter 70 to be transmitted to the payment meta-filter 76. Assume that specifications filtered by the high price filter 72 require either credit card 80 or cash 82 as payment and that a specification filtered by the low price filter can be processed by any of the three payment filters. QoS1 is filtered to produce QoS1*a* and QoS1*b*, while QoS2 is processed to produce QoS2*a*, QoS2*b*, and QoS2*c*. From a single preliminary QoS specification, five filtered multi-category QoS specifications emerge after filtering. It is evident from the example above that when one filter modifies a specification, the filter can modify the specification in such a manner that it may impose restrictions on the downstream filters such that some offers may be pruned out. Thus, interdependencies between the filters can be naturally captured through the pipeline design.

Filtering of a single preliminary QoS specification can generate multiple filtered QoS specifications. The potentially large quantity of filtered QoS specifications, hereinafter simply called QoS specifications or multi-category QoS specifications, poses a problem for data storage and transport if the QoS specifications are represented as individually distinct offers. In order to more efficiently store and transport the QoS specifications, a disjunctive representation can be used to represent the QoS specifications. Continuing with the example in FIG. 8, assume that QoS1*a* has had dimensions X and Y modified to Xa and Ya and that QoS1*b* has had dimensions X and Y modified to Xb and Yb. Assume further that both QoS1*a* and QoS1*b* have ten remaining dimensions in common. Instead of storing these identical ten dimensions twice for QoS1*a* and QoS1*b*, a single represen-

tation of these ten dimensions can be stored in one copy. Data can be stored in association with the single copy of the ten common dimensions and the X and Y dimensions which indicate the filtering steps performed on dimension X and Y to produce Xa and Ya and Xb and Yb. In this manner disjunctive representation of the QoS specifications enables efficient storage and transport.

Referring to FIG. 9, a schematic diagram of a disjunctive representation shows that the preliminary QoS specification is stored as a single copy. The filtered QoS specifications are represented by the modification steps which the filters in the filter pipeline execute. Continuing with the example presented in FIG. 8, the high price filter modification step is represented by the arrow labeled high. The credit card filter **80** and the cash filter **82** execute modification steps which are represented by the credit card arrow and the cash arrow. The low price filter modification step is represented by the low arrow which precedes the modification steps executed by the no payment filter **78**, the credit card filter **80** and the cash filter **82** represented respectively by the no payment arrow, the credit card arrow, and the cash arrow. In this manner a single copy of the preliminary QoS specification can be stored together with data representing the filtering steps performed on the preliminary QoS specification to produce the filtered QoS specifications.

The filters can also be configured to modify the preliminary QoS specifications to include out-of-band information such as implementation specific information, signal required capabilities, and parameterization. For instance, a filter might attach data which serves as a reminder to a particular implementation that certain resources must be available to execute a particular QoS agreement incorporating the terms of a particular QoS specification. A transaction identifier is also added to the preliminary specifications which becomes a deal identifier if the specification is incorporated into an agreement.

Returning to FIG. 7, in a preferred embodiment the server agent **40** includes an expected utility calculation program **54** which can be utilized to calculate the expected utility of offers received from the client agent **42**. A trust filter **48** enables the server agent to utilize trustworthiness data regarding the client to modify the expected utility of offers from the client agent by modifying the expected behavior data provided by a client agent according to the trustworthiness of the client agent. A QoS negotiation protocol **46** establishes a set of rules for transmission and receipt of messages via the signal transceiver **44** during the negotiation process with the client agent **42**.

The client agent **42** is associated with a client application **66** which can be utilized to create preliminary QoS specifications to be included in an offer for a QoS agreement. A client implementation filter pipeline **65** modifies the preliminary QoS specifications according to general resource capabilities of the system. The client agent **42** further includes an expected utility calculation program **64** and a QoS specification selection program **62** for selection of QoS specifications to be included into offers to the server agent **40**. A client trust filter **60** modifies QoS specifications received from the server agent **40** according to trustworthiness data for the server agent **40**. The client agent further includes a QoS negotiation protocol **58** and a signal transceiver **56** for communication with the server agent **40** during negotiation of a QoS agreement.

Referring to FIGS. **7** and **10**, a method for the operation of a server agent **40** during a QoS negotiation includes the step **84** of a server-associated application **52** generating a preliminary QoS specification. As previously discussed, the server application **52** does not take into account the limitations of its environment when generating the preliminary specification. In step **86** the preliminary QoS specification is modified by the implementation filters **50** to produce QoS specifications which reflect resource limitations of the distributed system. The client agent **42** requests a set of offers which include QoS specifications from the server agent **40**, and in step **88** the server agent transmits a set of QoS specifications to the client agent **42**. The offer transmitted by the server agent **40** includes a QoS specification and an associated client behavior profile which the server agent **40** requires from the client in order to provide the QoS indicated in the QoS offer.

As will be discussed in greater detail below, the client agent **42** performs conformance checking and expected utility calculations on the QoS specifications transmitted by the server agent **40**. In step **90**, the server agent **40** receives an offer which might contain a QoS specification and an associated client behavior profile, which the client agent **42** proposes as a basis for a QoS agreement. Alternatively, if none of the QoS specifications which the server agent **40** transmitted were deemed to be satisfactory, the client agent **42** might transmit a conflict deal message indicating it has no desire to negotiate any further.

If the offer received in step **90** proposes a QoS agreement, the server agent **40** calculates an expected utility for the offer in step **98**. The utility calculation might be based on a server operating mode preference function and/or a client behavior profile utility function. The server operating mode preference function correlates particular QoS specifications to utility values to the server and the client behavior mode preference function correlates the client behavior profiles to expected utilities to the server. Based on the utility calculation, in step **100** the server agent selects an offer to be considered for acceptance.

In a preferred mode, the server agent **40** employs a trust function in the utility calculation process. The trust function is representative of a level of trust which the server agent **40** possesses with regard to the client agent and it is used to modify values in the client's offer resulting in a modification of the calculated expected utility. For instance, the client agent **42** might represent that the client will make no more than ten calls in any 24-hour period under the proposed agreement. However, the server agent **40** might have information which indicates that the client has broken that same promise in the past 90% of the time. Consequently, even if the server agent would have accepted the client agent's offer if the promise were taken at face value, upon filtering the offer using the utility function, the server agent **40** might reject the offer. The client agent trustworthiness data which the trust function utilizes can be based on past interactions the server agent **40** has had with the client agent **42**, or the data can be obtained from a reputation broker, not shown. The reputation broker has either polled the trustworthiness data from other servers or has entered into QoS agreements itself with the client agent to determine the client agent's trustworthiness. A trust function can also be utilized to filter information provided by the reputation broker based on the trustworthiness of the broker.

In step **102** the server agent determines if sufficient resources are available to the server to support the requirements of the selected offer. Unlike the filtering of the application-generated preliminary QoS specification, the determination of resource availability in step **102** involves specifically determining if previous commitments assigned to the resources necessary to support the proposed QoS

levels prevent the server from being able to fulfill the terms of the proposed agreement. If it is determined that sufficient resources are available, the agent accepts the offer in step **104**. If sufficient resources are not available, in step **106** it is determined whether any alternative offers are available for consideration. If no alternative offers are available, the server agent transmits a conflict deal message in step **108**. Alternatively, the server agent might transmit a counteroffer in step **108**.

With reference to FIGS. **7** and **11**, a method for utilizing a client agent **42** in a QoS negotiation includes the step of transmitting a request for an offer sent to a server agent **40** in step **110**. In step **112** an offer set is received from the server agent **40** which includes multiple QoS specifications. A possibility exists that some of the QoS specifications included in the offer set do not satisfy minimum requirements as expressed by a constraint profile. Because calculation of expected utility is a resource intensive operation, it is not desirable to perform a full expected utility calculation on a particular QoS specification if that QoS specification does not satisfy the minimum requirements. The client agent **42** performs a conformance checking operation in step **114**, wherein the client constraint profile is referenced to determine in step **116** if the QoS specification satisfies the minimum requirements of the client. If the particular QoS specification is determined not to satisfy the minimum requirements, a determination is made whether any other offers are to be considered by the client agent **42**. If additional offers are available, then the client agent returns to step **114** to perform conformance checking on the remaining offers.

Prior to commencing the utility computation process for any offers which satisfy the minimum requirements, the contracts for each operation in the QoS specification are identified. Those contracts which do not have a corresponding contract in the constraint profile are omitted from the calculation process. Any contracts in the QoS specification which are not found in the client's constraint profile involve QoS guarantees which the client does not require.

If an offer is determined to satisfy the minimum requirements in step **116**, a utility profile is utilized to establish which utility functions are to be utilized in calculating the expected utility of dimensions. Referring back to the example in FIG. **6**, the utility function for availability establishes a utility distribution in step **118** which correlates the values for availability to utility values to the client. In step **120** the dimension values specified in the QoS specification (a, b) are mapped to utility values and the expected utility value can be calculated by, for example, calculating the mean utility in the interval between the availability values a and b.

In step **122** the expected utilities of the contracts and operations are calculated to arrive at an expected utility of the profile, that is the QoS specification. The corresponding contract utility function is applied to each of the contracts to obtain the utility of each contract. In a preferred mode, the utility of each contract is normalized by dividing the utility by the sum of the utilities of each dimension defined in the contract to obtain the expected utility of each contract. The aggregated utility of an operation is calculated by multiplying the weights of each contract, as specified by the operation utility function, with the calculated expected utility of each contract and summing the results. The expected utility of an operation is then calculated by dividing its utility by the sum of the weights of all the contracts defined for the operation. The expected utility of each operation is multiplied by the weight associated with the operation. The aggregated utility of a QoS specification is the sum of all the weighted utilities of the operations. The expected utility of a specification is obtained by dividing its utility by the sum of the weights of all the operations defined for the specification.

A more formal description of the algorithm for calculating the expected utility is given in the code below. SP stands for server profile, CP for constraint profile, EW stands for expected utility of a server profile, wf_c stands for the utility function that calculates the utility of a contract in an operation, wf_p stands for the utility function that returns the weight of an operation. The function Sums (X) returns the sum of the weights and values of all Xs. That is, if X=C, then Sums (X) returns the sum of the values of every dimension in every contract. If X=0, then the function returns the sum of the weights for all of the contracts. If X=P, then Sums (P) returns the sum of the weights of all the operations.

```
Begin
 EW = 0; Utility_p = 0;
 For all operations O in SP
 {
  Expect_o = 0; Utility_o = 0;
  For all contracts C for O in SP and CP
  {
    Expect_c = wf_c(C) / Sums(C);
    Utility_o += wf_o(C) * Expect_c;
  }
  Expect_o = Utility_o / Sums(O);
  Utility_p += wf_p(O) * Expect_o;
 }
 EW = Utility_p / Sums(P);
End;
```

The client agent **42** also preferably utilizes a trust function in the calculation of expected utility of QoS specifications. The client agent's use of the trust function is analogous to that of the server agent **40**. The trust function is utilized to modify declared values of dimension to reflect the believed value to the client agent. For example, if the value for availability specified in a QoS specification as having a 90% probability of providing 98% availability, the trust function might reduce the stated probability to a believed probability of 70%. Again, the trust function can be based upon trustworthiness data for the server agent **40** based on past interactions with the server agent. Alternatively, the server agent trustworthiness data is assembled by a reputation broker and transmitted to the client agent upon a request for the trustworthiness data.

In step **124**, the client agent **42** selects at least one QoS specification to be included into an offer for a QoS agreement. In step **126** the client agent **42** transmits the offer to the server agent **40** and in step **128**, the client agent monitors the connection to the server agent for either an acceptance, a rejection, or a counteroffer.

A negotiation protocol establishes the rules for transmission of messages which comprise the communication between the server agent **40** and the client agent **42** during the negotiation process. The messages that can be exchanged between two negotiating agents can be informally described as follows:

request offer set: Sent by client to request all the offers that the server is willing to accept in a deal. The message conveys a client profile.

send offer set: Sent by the server in response to a request offer set message. The message body contains all the offers that the server supports. The client may select one of these and propose it as a deal to the server.

send offer: Sent by either agent to make an offer (or counteroffer) to the other. It can be sent as a response to a send offer.

deal: Sent by an agent that received an acceptable offer from the other agent. The message indicates that the agent that received the most recent offer is willing to accept it as a deal.

acknowledge deal: Sent by either agent as a response to the deal message.

conflict deal: Sent by either agent as a response to a send offer message. The message indicates that the received offer was not acceptable and that the agent does not intend to make a counteroffer.

acknowledge conflict deal: Sent by either agent as a response to a conflict deal.

stop: This message can be sent by either agent to stop negotiating immediately. It can be due to a failure, application shutdown, change of priorities or any other reason.

The message send offer conveys an offer consisting of a server profile and a client profile. The message send offer set carries a set of offers provided by the server. All of the messages convey a transaction identifier, that becomes the deal identifier if a deal is reached. The deal identifier is used to associate individual operation invocations with deals.

The set of messages sent from the application to the negotiation mechanism can be summarized as follows:

abort: Sent by the application to its negotiation mechanism to indicate the termination of all ongoing negotiations for that application.

neg: Sent by the application to initiation negotiation.

regOffers: Sent by server application to register new any offers that provides.

FIGS. **12** and **13** illustrate the client-side and server-side negotiation state machines respectively. The procedures for the client and server negotiation mechanism are illustrated as state-charts. A box represents a state and an arrow a state transition. Transitions are represented by labels with zero or one incoming or outgoing events and zero or more actions. Incoming events, outgoing events, and actions are represented as follows: Event/^out event/action. States can be nested and a transition from an outer state indicates a transition that applies to all inner states. A circle represents a start state, and two nested circles represent a valid end state.

Referring to FIG. **12**, a client negotiation mechanism is normally in an idle state **130**. When it receives a negotiation request from the client, the client mechanism leaves the idle state **130** and enters into a negotiate state **131**. The client negotiation mechanism also transitions to an initialOffer state **132** in response to the negotiation request. In initialOffer **132**, the mechanism determines if its server offers are valid or if it requires an updated set of server offers. If an updated set of offers is required, the client mechanism transitions to the requestMode **134**, wherein the client mechanism transmits a request offer set message to the server agent **40**. If the client mechanism already has a valid set of server offers, it transitions to the computeOffer state **136**, wherein the client mechanism computes an offer to transmit to the server agent **40**. If the offer is for a conflict deal, the mechanism transitions to a notOk state **152**. If the client mechanism receives an ackConflictDeal message in the notOK state **152**, the client mechanism transitions to a noDeal end state **158**.

If the client mechanism transmits an offer for a QoS agreement at state **136**, it transitions to the waitForResp state **142**. The client mechanism can receive three different messages in the waitForResp state **142**, a deal message, a conflictDeal message, and a sendOffer message. The deal message indicates acceptance from the server negotiation message. In response, the client mechanism transitions to the okayed state **140**, wherein it transmits an acknowledge deal message. The client mechanism then transitions to deal end state **156**. If the client mechanism receives a conflict deal message while in the waitForResp state **142**, the mechanism transitions to notOkayed **150**. If the client mechanism receives an acknowledge conflict deal in the notOkayed state **150**, it transitions to noDeal **158**. If the mechanism receives a sendOffer message in the waitForResp state **142**, the client mechanism transitions to an incomingOff state **138**. If while in the incomingOff state **138** the client mechanism receives a conflict deal message, it transitions to the notOk state **152**. If the client mechanism transmits a deal message while in the incomingOff state **138**, it transitions to the ok state **148**, wherein it receives an acknowledge deal message to transition to the deal end state **156**. The negotiation will always be interrupted if the client mechanism receives the stop message. If the client negotiation mechanism transmits a stop message, it transitions to the abort state **154** and the negotiation is interrupted.

Referring to FIG. **13**, the server-side negotiation mechanism transitions from an idle state **160** to a negotiate state **161** upon receiving either an offer or an offer request. If the server negotiation mechanism receives an offer, it transitions to an evalOffer state **162**. In the evalOffer state the server mechanism can either transmit a conflict deal message to arrive at the notOk state **176** or a deal message to arrive at the ok state **178**. If the client mechanism transmits an acknowledge conflict deal while the server mechanism is in the notOk state **176**, the server mechanism transitions to the noDeal end state **184**. If the server mechanism receives an acknowledge deal while in the ok state, it transitions to the deal end state **182**. If the server mechanism desires to transmit a counteroffer while in the evalOffer state **162**, it transitions to the computeOffer state **170**. The server mechanism transmits the counteroffer in a send offer message and transitions to a waitForResp state **168**.

While in the waitForResp state **168**, the server negotiation mechanism can receive a deal message or a conflict deal message. Receiving the conflict deal message causes the server mechanism to transition to the notOkayed state **174** from which it transmits an acknowledge conflict deal message to arrive at the noDeal end state **184**. Receiving the deal message while in the waitForResp state **168** causes the server mechanism to transition to the okayed state **172** from which the mechanism transmits an acknowledge deal message to arrive at the deal end state **182**.

If the server mechanism receives a request offer set message while in the idle state **160**, it transitions to a collectOffs state **164**. The server mechanism transmits a send offer set message and transitions to waitForOff state **166** from in which the server mechanism will either receive a conflict deal message or a send offer message which includes an offer for a QoS agreement. If the server mechanism receives a conflict deal message while in the waitForOff state **166**, it transitions to the notOk state **176** from which the server mechanism transmits an acknowledge conflict deal message to arrive at the noDeal state **184**. If the server mechanism receives an offer for a QoS agreement while in the waitForOff state **166**, the mechanism transitions to the evalOffer state **162**.

If, while in the negotiate state **161**, the server negotiation mechanism receives an abort message or transmits a stop

message, the server mechanism transitions to an abort state **180** and the negotiation is interrupted. If the server negotiation message receives a stop message while in the negotiation state **161**, it transitions to the idle state **160**.

What is claimed is:

1. A method for negotiating a multi-category quality-of-service (QoS) agreement between a client and a server in a system comprising the steps of:
   establishing a communications link having a client-side of said negotiating and server-side;
   generating a plurality of multi-category QoS specifications, each said multi-category QoS specification being representative of a probabilistic estimate of QoS levels for a service available via said server, each category of said QoS specifications representing a different QoS-related parameter;
   transmitting said plurality of multi-category QoS specifications over said communications link from said server-side to said client-side;
   selecting at least one of said multi-category QoS specifications to be included in an offer for a QoS agreement;
   identifying each said selected multi-category QoS specification as part of said offer, said offer requesting a service provided by said server at said probabilistic estimate of said QoS levels represented by said each selected multi-category QoS specification;
   transmitting said offer over said communications link from said client-side to said server-side; and
   monitoring said communications link at said client-side for a response to said offer indicating one of an acceptance, a rejection, and a counteroffer to said offer.

2. The method of claim 1 further comprising the step of calculating an expected utility to said client for each said QoS specification, including calculating expected utilities of categories of said each multi-category QoS specification received via said communications link, said calculation of said expected utilities being performed at said client-side of said communications link and being based on said probabilistic estimate of said QoS levels available via said server, and further, wherein said step of selecting said at least one multi-category QoS specification includes basing said selection on said calculation of said expected utility.

3. The method of claim 2 wherein expected utility is specific to relative significance to said client and wherein said calculating step for each said multi-category QoS specification includes a hierarchical approach of:
   establishing low level expected utilities for each of a plurality of dimensions associated with different categories of said multi-category QoS specifications, each dimension being representative of one of a qualitative and a quantitative QoS-related parameter of a category with which said each dimension is associated;
   establishing an intermediate level expected utility for each said category based upon said low level expected utilities established for dimensions associated with said each category, said intermediate level expected utilities being said expected utilities of said categories; and
   establishing a high level expected utility for each said multi-category QoS specification based upon said intermediate level expected utilities established for said each multi-category QoS specification, said high level expected utility being said expected utility of said each multi-category QoS specification.

4. The method of claim 2 further comprising the step of:
   determining whether said expected utility of each said multi-category QoS specification satisfies a preselected minimum expected utility threshold; and
   if an expected utility of a particular multi-category QoS specification does not satisfy said minimum expected utility threshold, disabling consideration of said particular multi-category QoS specification from inclusion into said offer.

5. The method of claim 2 further comprising the step of referencing a constraint profile which defines minimum requirements of said client to determine whether to perform said step of calculating said expected utility.

6. The method of claim 2 wherein said step of calculating said expected utility includes filtering said each multi-category QoS specification utilizing a trust function representative of a level of trust said client has with respect to said server, said trust level being indicative of a degree to which said server has satisfactorily performed under terms of past QoS agreements.

7. The method of claim 1 further comprising the steps of:
   generating a preliminary multi-category QoS specification utilizing an application-associated with said server; and
   filtering said preliminary multi-category QoS specification to modify said preliminary multi-category QoS specification according to a capability of said system to support preliminary QoS levels represented by said preliminary multi-category QoS specification, said filtering of said preliminary multi-category QoS specification generating said at least one multi-category QoS specification.

8. The method of claim 1 further comprising the steps of:
   receiving said offer that includes said at least one selected multi-category QoS specification;
   determining whether said system is capable of supporting a QoS level associated with said selected multi-category QoS specification, said determination being based at least partially on previous commitments assigned to said system; and
   deciding whether to accept said offer on behalf of said server based at least partially on said determination of whether said system is capable of supporting said QoS level.

9. A system for enabling multi-category QoS negotiation toward a multi-category QoS agreement between a server and a client in a system comprising:
   a server agent comprising:
      a) means for establishing multi-category QoS specifications representative of probabilistic estimates of QoS levels associated with alternative services provided by said server;
      b) first means for transmitting a plurality of said multi-category QoS specifications in response to receiving a multi-category QoS specification request for communication involving said client; and
      c) means for determining whether to enable said services based upon multi-category QoS offers received on behalf of said server; and
   a client agent comprising:
      a) means, responsive to said first means to receive said plurality of multi-category QoS specifications, for selecting one of said multi-category QoS specifications for identification to said server agent on behalf of said client; and
      b) second means for identifying said selected multi-category QoS specification to said determining means of said server agent as at least a portion of a selected QoS offer.

10. The system of claim 9 wherein said client agent further includes means, connected to said first means and

said selecting means, for calculating an expected utility to said client for each of said multi-category QoS specifications based on said probabilistic estimates of said QoS levels, and further, wherein said selecting means is responsive to said expected utility calculation by said calculating means.

11. The system of claim 9 wherein said probabilistic estimates of said QoS specifications are expressed for said categories of said multi-category QoS specifications in the absence of a linear combination of said probabilistic estimates.

12. The system of claim 10 wherein said calculating means of said client agent includes a client-agent trust filter configured to modify data within said multi-category QoS specification according to a first trust function, said first trust function being representative of a level of trust possessed by said client agent with respect to said server agent based on an extent to which QoS agreements negotiated by said server agent have been satisfactorily performed in the past, said determining means of said server agent including a server-agent trust filter configured to modify data within said selected multi-category QoS offer according to a second trust function, said second trust function being representative of a level of trust possessed by said server agent with respect to said client agent based on an extent to which QoS agreements negotiated by said client agent have been satisfactorily performed in the past.

13. The system of claim 10 wherein said calculating means of said client agent includes an expected default provision function enabled to calculate expected utilities for dimensions of said categories not included in said multi-category QoS specifications.

14. The system of claim 9 wherein said server agent further includes:

means, connected to said determining means of said server agent, for ascertaining whether sufficient resources are available to said server to support resource requirements of said selected multi-category QoS offer; and

means, responsive to said determining means and said ascertaining means, for selecting a second multi-category QoS specification for inclusion into a counteroffer in response to a determination that said resource requirements are unavailable to said server.

15. The system of claim 9 wherein said establishing means of said server agent includes an application enabled to generate a preliminary multi-category QoS specification, said establishing means further including a plurality of filters associated with resources of said system supporting said service provided by said server, said filters being configured to modify said preliminary multi-category QoS specification to reflect capabilities of said resources to support said service, said modification of said preliminary multi-category QoS specification resulting in at least one of said multi-category QoS specifications.

16. The system of claim 9 wherein said first means for transmitting and said second means for identifying both include a negotiation protocol having rules for transmission of messages during said multi-category QoS negotiation.

17. A method for negotiation of a multi-category QoS deal between a server and a client in a distributed system comprising the steps of:

generating an initial multi-category QoS specification representative of probabilistic estimates of QoS levels available via said server based on preferences by a server application;

filtering said initial multi-category QoS specification according to capabilities of resources of said distributed system to support QoS levels, said filtering producing a plurality of multi-category QoS specifications, each category of said multi-category QoS specifications being representative of a parameter of said multi-category QoS specification;

transmitting said plurality of multi-category QoS specifications to a client agent in response to a request via a communications link;

upon receiving said plurality of multi-category QoS specifications at said client agent, calculating an expected utility at said client agent for each of said plurality of multi-category QoS specifications based on said probability estimates of said QoS levels;

formulating an offer including at least one of said multi-category QoS specifications in response to said calculations of expected utilities;

identifying said offer to a server agent; and

upon receiving said offer at said server agent, determining whether to accept said offer, including utilizing a first trust function that is representative of a level of trust said server agent possesses with regard to said client agent based on an extent to which QoS agreements negotiated by said client agent have been satisfactorily performed previously, said calculating step including utilizing a second trust function in calculating said expected utility of said multi-category QoS specification, said second trust function being representative of a level of trust said client agent possesses with regard to said server agent based on an extent to which QoS agreements negotiated by said server agent have been satisfactorily performed previously.

18. The method of claim 17 wherein said determining step includes ranking said at least one selected multi-category QoS specification according to an expected utility to said server of each of said at least one selected multi-category QoS specification in said offer.

19. The method of claim 17 wherein said filtering step includes producing said plurality of multi-category QoS specifications such that each multi-category QoS specification includes a client behavior mode with a probabilistic estimate of client behavior.

* * * * *

US006556565B1

(12) United States Patent
Ward et al.

(10) Patent No.: US 6,556,565 B1
(45) Date of Patent: Apr. 29, 2003

(54) INTERNET PROTOCOL (IP) TELECOMMUNICATION

(75) Inventors: David P. Ward, Belleville (CA); Cuthbert Cheung, Belleville (CA); John Marshall, Belleville (CA); Ian Astle, Amherstview (CA)

(73) Assignee: Nortel Networks Limited, St. Laurent (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: 09/247,915

(22) Filed: Feb. 11, 1999

Related U.S. Application Data

(60) Provisional application No. 60/091,457, filed on Jul. 1, 1998.

(51) Int. Cl.[7] H04L 12/66

(52) U.S. Cl. 370/356; 370/493

(58) Field of Search 370/493, 356, 370/261, 466, 395, 468, 469, 471, 230, 271, 401, 236, 352; 709/226, 223, 224; 379/114, 231

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,761,294	A	*	6/1998	Shaffer et al.	379/230
5,764,645	A	*	6/1998	Bernet et al.	370/466
5,892,764	A	*	4/1999	Riemann et al.	370/401
5,944,795	A	*	8/1999	Civanlar	709/227
6,032,272	A	*	2/2000	Soirinsuo et al.	714/706
6,044,081	A	*	3/2000	Bell et al.	370/401
6,137,869	A	*	10/2000	Voit et al.	379/114
6,154,776	A	*	11/2000	Martin	709/226
6,175,871	B1	*	1/2001	Schuster et al.	709/231
6,282,192	B1	*	8/2001	Murphy et al.	370/352
6,363,065	B1	*	3/2002	Thornton et al.	370/352

FOREIGN PATENT DOCUMENTS

EP	1111859 A2	*	11/2000	370/352

OTHER PUBLICATIONS

Thomas A., et al, "Asynchronous Time–Division Techniques: An Experimental Packet Network Intergrating Videocommunication", International Switching Symposium (ISS), Florence, Italy, May 7–11, 1984, Session 32C, Paper 2, pp. 1–7.

Coundreuse, J–P., et al, "Prelude: An Asynchronous Time–Division Switched Network", IEEE International Conference on Communications, Jun. 9, 1987, vol. 2 of 3, pp. 0769–0773.

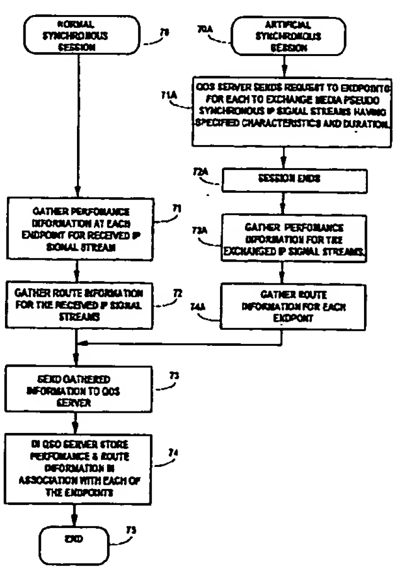
* cited by examiner

Primary Examiner—Hassan Kizou
Assistant Examiner—Anh-Vu H Ly
(74) *Attorney, Agent, or Firm*—Nortel Networks

(57) **ABSTRACT**

A quality of service (QOS) server gathers information in relation to transport introduced delays and obliterations of so called real-time protocol (RTP) signal streams which are used to packetize signals of synchronous signals origins, for example origins such as telephones and raster scan apparatus, for transport via the IP network. The QOS server is able to provide recently gathered QOS information about a requested call, apriori the actual call set up. In an example of a PBX call centre, an automatic dialler is coupled with a QOS server and in relation to endpoint and route selected data, determines if an IP telephony call should proceed, contingent upon the probable QOS being sufficient, before a call controller initiates a call set up.

21 Claims, 7 Drawing Sheets



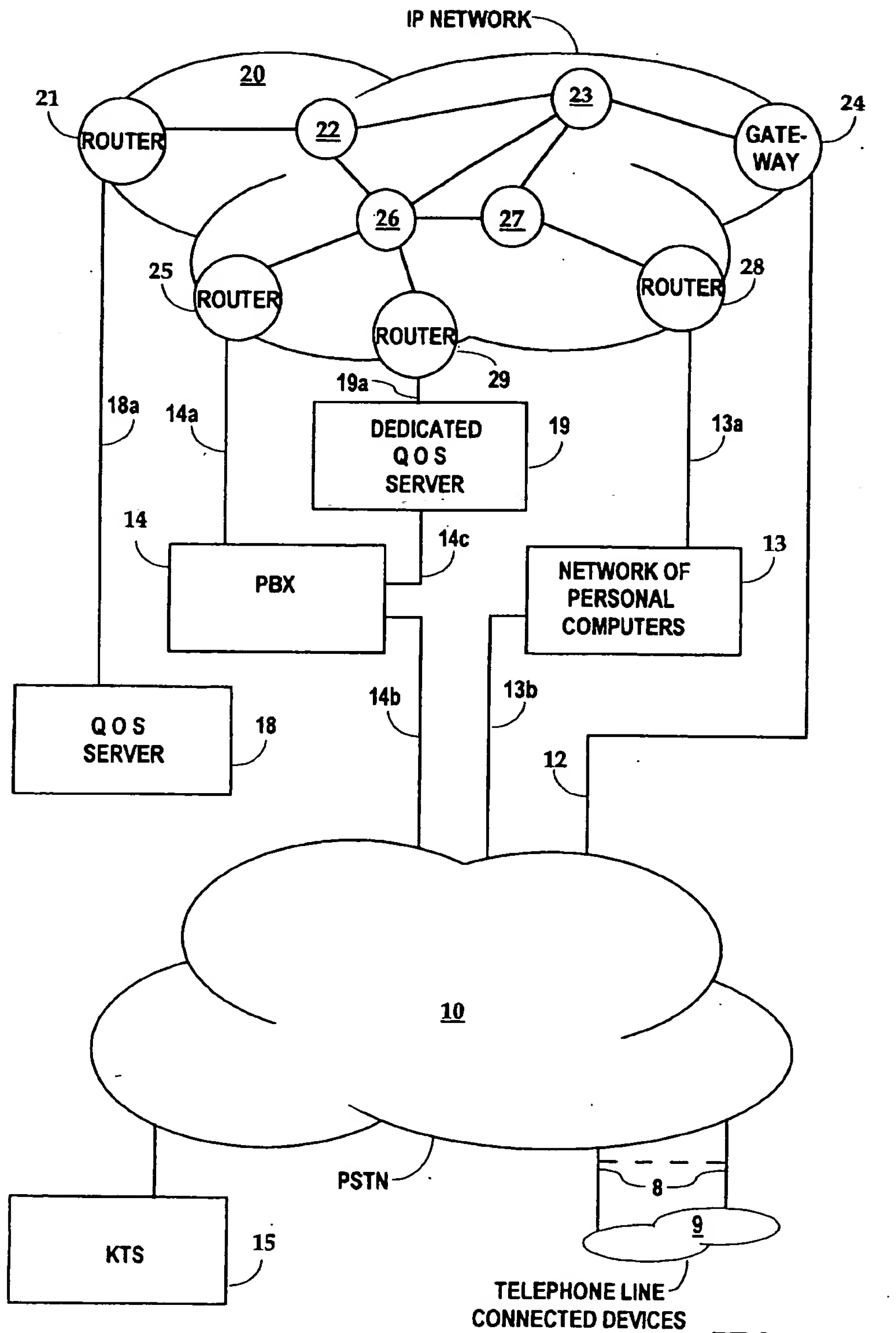

IP NETWORK
20
21
ROUTER
22
23
24
GATE-WAY
25
ROUTER
26
27
ROUTER
28
ROUTER
29
19a
18a
14a
DEDICATED QOS SERVER
19
13a
14
PBX
14c
NETWORK OF PERSONAL COMPUTERS
13
QOS SERVER
18
14b
13b
12
10
PSTN
8
9
KTS
15
TELEPHONE LINE CONNECTED DEVICES

Fig. 1

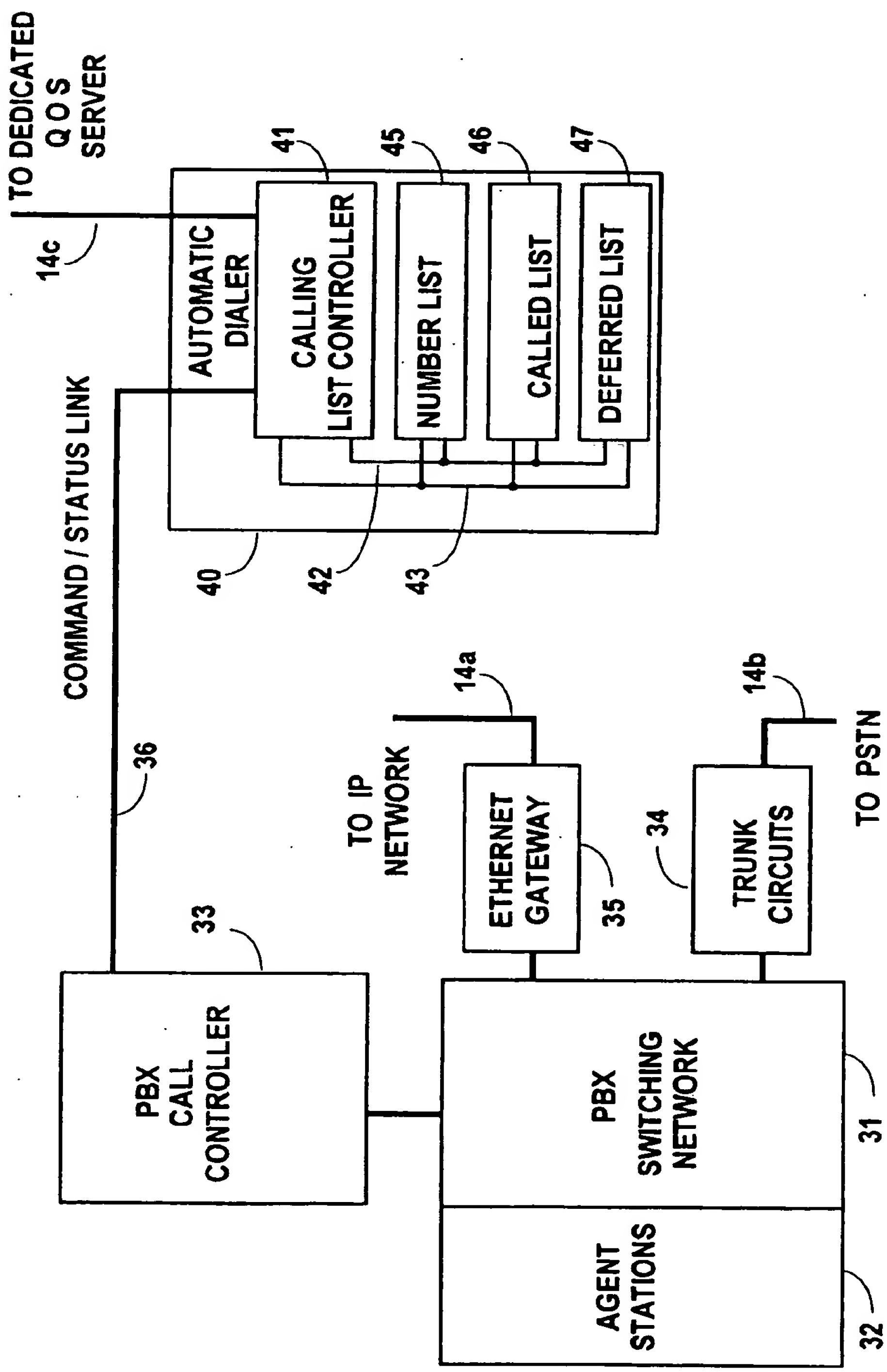

TO DEDICATED
Q O S
SERVER
14c
AUTOMATIC
DIALER
CALLING
LIST CONTROLLER
NUMBER LIST
CALLED LIST
DEFERRED LIST
41
45
46
47
40
42
43
COMMAND / STATUS LINK
36
14a
TO IP
NETWORK
ETHERNET
GATEWAY
35
14b
TO PSTN
TRUNK
CIRCUITS
34
33
PBX
CALL
CONTROLLER
PBX
SWITCHING
NETWORK
31
AGENT
STATIONS
32

Fig. 2

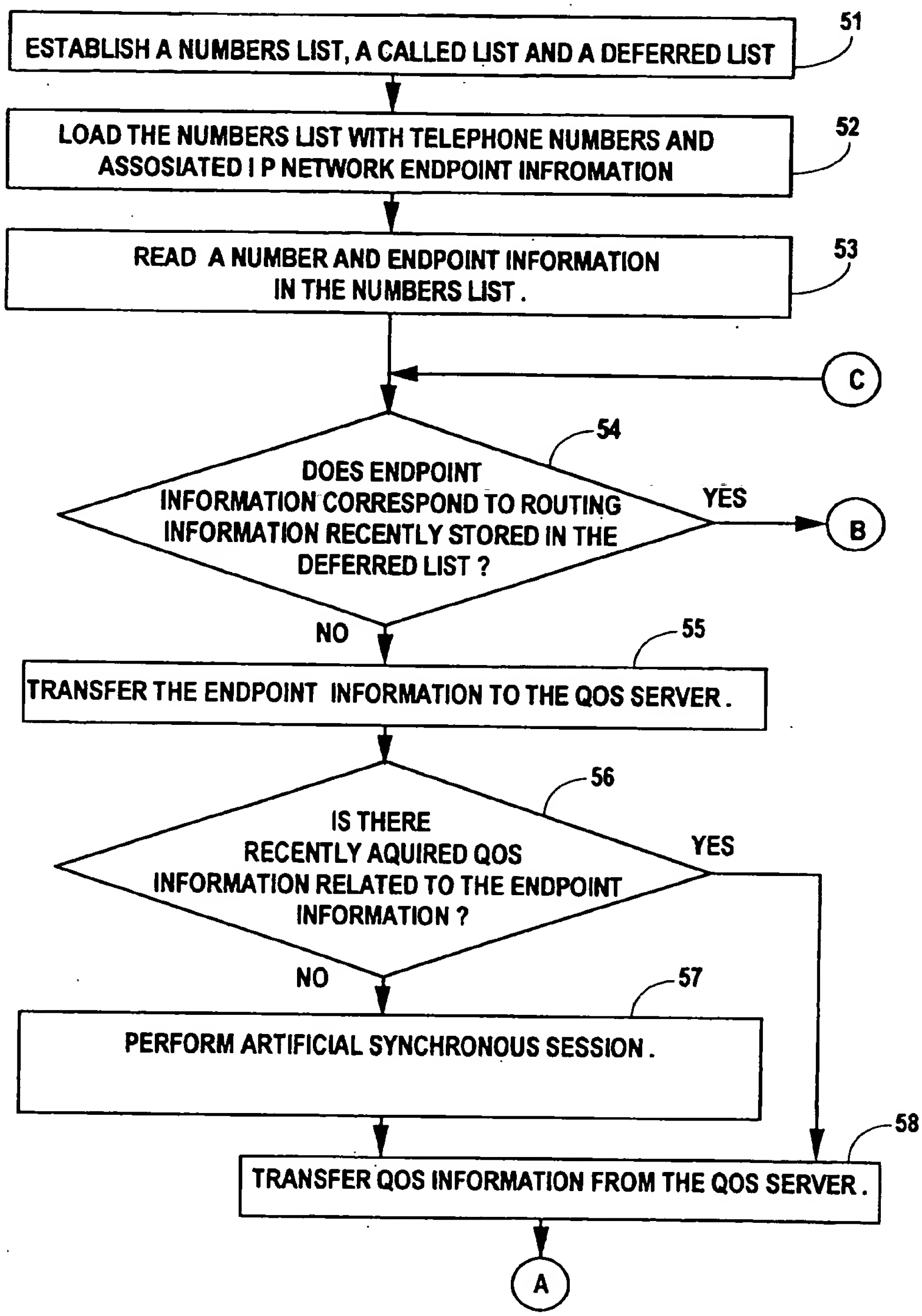

ESTABLISH A NUMBERS LIST, A CALLED LIST AND A DEFERRED LIST
51
LOAD THE NUMBERS LIST WITH TELEPHONE NUMBERS AND ASSOSIATED I P NETWORK ENDPOINT INFROMATION
52
READ A NUMBER AND ENDPOINT INFORMATION IN THE NUMBERS LIST.
53
C
DOES ENDPOINT INFORMATION CORRESPOND TO ROUTING INFORMATION RECENTLY STORED IN THE DEFERRED LIST ?
54
YES
B
NO
TRANSFER THE ENDPOINT INFORMATION TO THE QOS SERVER.
55
IS THERE RECENTLY AQUIRED QOS INFORMATION RELATED TO THE ENDPOINT INFORMATION ?
56
YES
NO
PERFORM ARTIFICIAL SYNCHRONOUS SESSION.
57
TRANSFER QOS INFORMATION FROM THE QOS SERVER.
58
A

Fig. 3

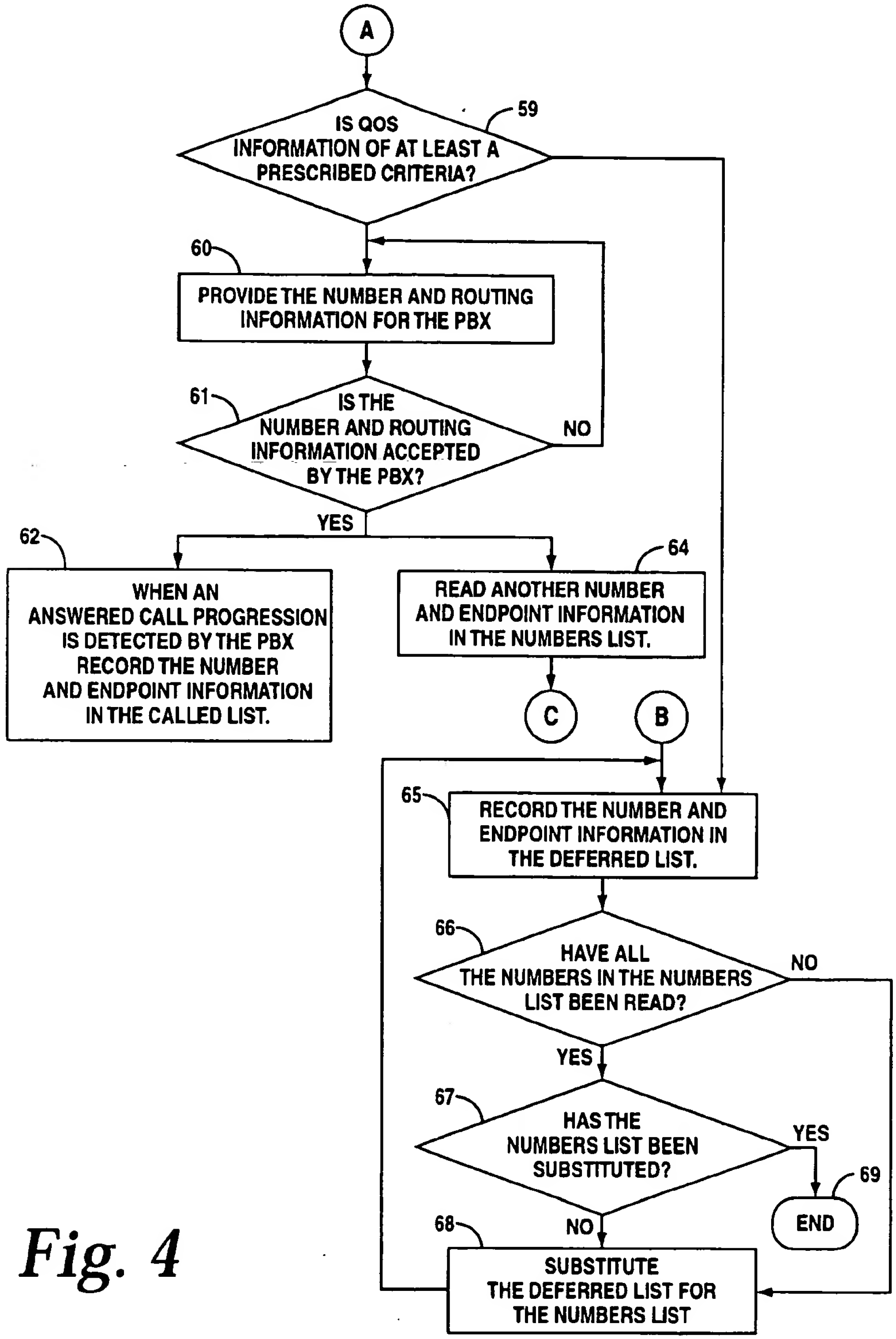

A
59
IS QOS INFORMATION OF AT LEAST A PRESCRIBED CRITERIA?
60
PROVIDE THE NUMBER AND ROUTING INFORMATION FOR THE PBX
61
IS THE NUMBER AND ROUTING INFORMATION ACCEPTED BY THE PBX?
NO
YES
62
WHEN AN ANSWERED CALL PROGRESSION IS DETECTED BY THE PBX RECORD THE NUMBER AND ENDPOINT INFORMATION IN THE CALLED LIST.
64
READ ANOTHER NUMBER AND ENDPOINT INFORMATION IN THE NUMBERS LIST.
C
B
65
RECORD THE NUMBER AND ENDPOINT INFORMATION IN THE DEFERRED LIST.
66
HAVE ALL THE NUMBERS IN THE NUMBERS LIST BEEN READ?
NO
YES
67
HAS THE NUMBERS LIST BEEN SUBSTITUTED?
YES
69
END
68
NO
SUBSTITUTE THE DEFERRED LIST FOR THE NUMBERS LIST

Fig. 4

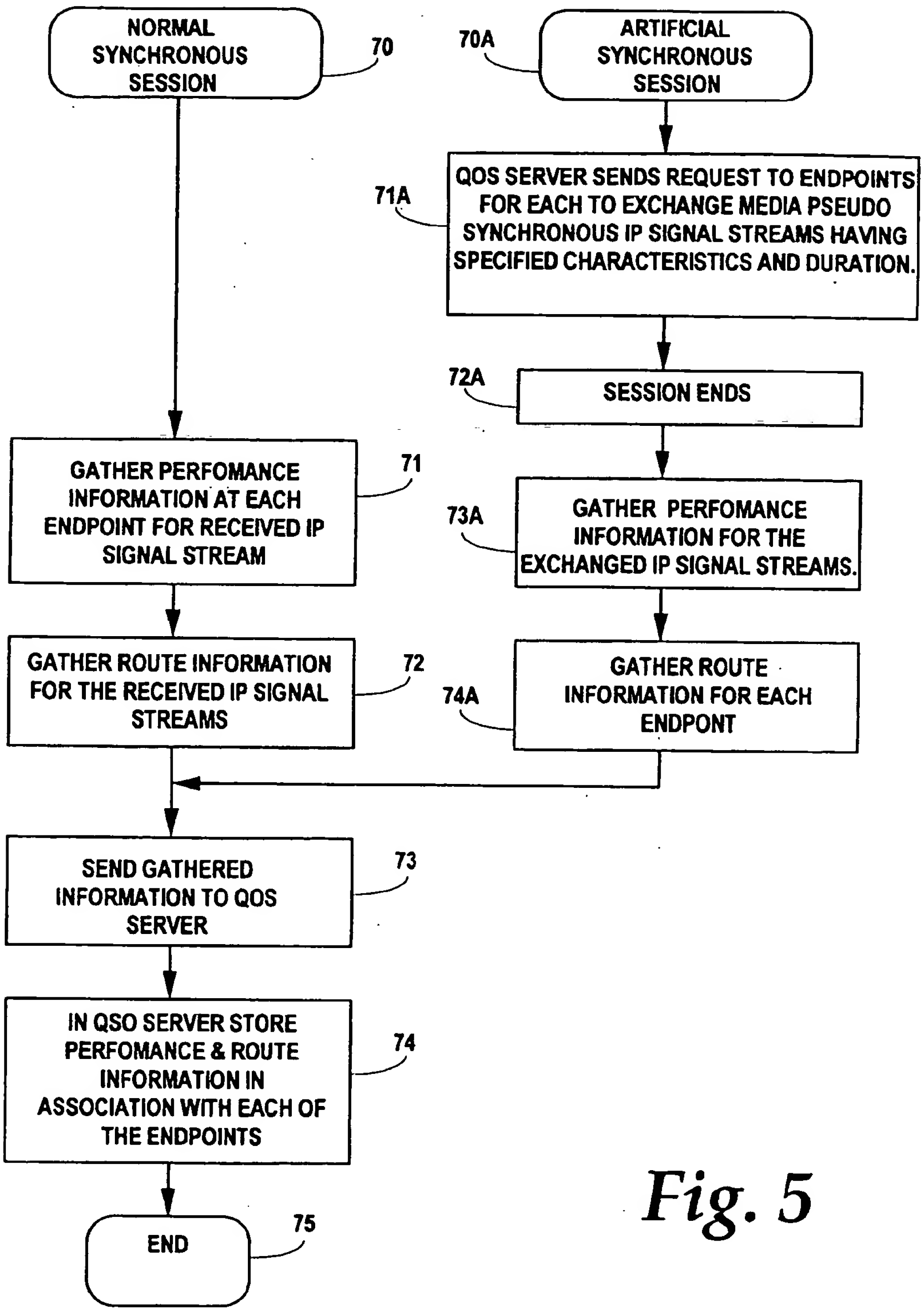

NORMAL SYNCHRONOUS SESSION
70
70A
ARTIFICIAL SYNCHRONOUS SESSION
QOS SERVER SENDS REQUEST TO ENDPOINTS FOR EACH TO EXCHANGE MEDIA PSEUDO SYNCHRONOUS IP SIGNAL STREAMS HAVING SPECIFIED CHARACTERISTICS AND DURATION.
71A
72A
SESSION ENDS
GATHER PERFOMANCE INFORMATION AT EACH ENDPOINT FOR RECEIVED IP SIGNAL STREAM
71
73A
GATHER PERFOMANCE INFORMATION FOR THE EXCHANGED IP SIGNAL STREAMS.
GATHER ROUTE INFORMATION FOR THE RECEIVED IP SIGNAL STREAMS
72
74A
GATHER ROUTE INFORMATION FOR EACH ENDPONT
SEND GATHERED INFORMATION TO QOS SERVER
73
IN QSO SERVER STORE PERFOMANCE & ROUTE INFORMATION IN ASSOCIATION WITH EACH OF THE ENDPOINTS
74
END
75

Fig. 5

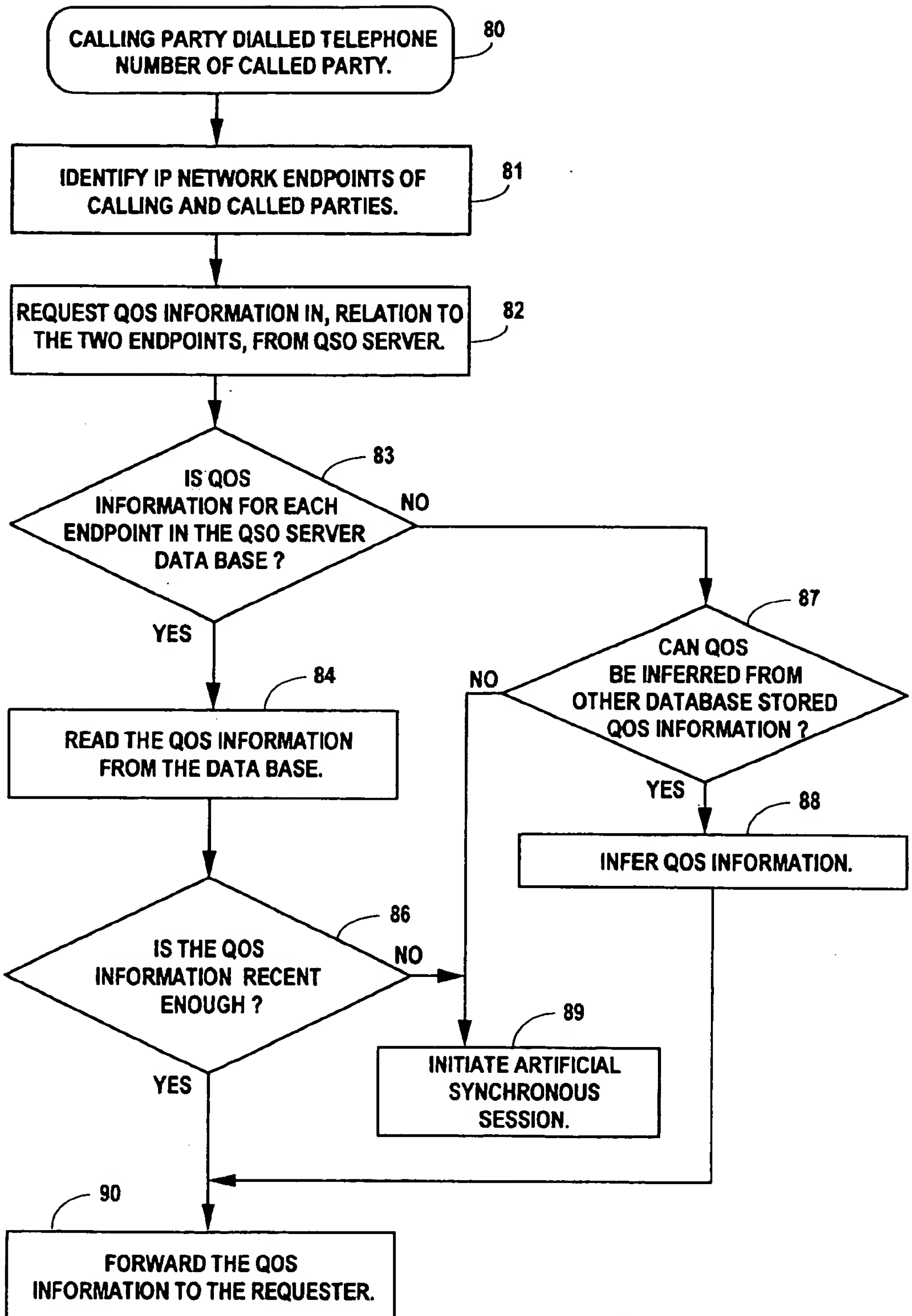

CALLING PARTY DIALLED TELEPHONE NUMBER OF CALLED PARTY.
80
IDENTIFY IP NETWORK ENDPOINTS OF CALLING AND CALLED PARTIES.
81
REQUEST QOS INFORMATION IN, RELATION TO THE TWO ENDPOINTS, FROM QSO SERVER.
82
IS QOS INFORMATION FOR EACH ENDPOINT IN THE QSO SERVER DATA BASE ?
83
NO
YES
READ THE QOS INFORMATION FROM THE DATA BASE.
84
CAN QOS BE INFERRED FROM OTHER DATABASE STORED QOS INFORMATION ?
87
NO
YES
INFER QOS INFORMATION.
88
IS THE QOS INFORMATION RECENT ENOUGH ?
86
NO
YES
INITIATE ARTIFICIAL SYNCHRONOUS SESSION.
89
FORWARD THE QOS INFORMATION TO THE REQUESTER.
90

Fig. 6

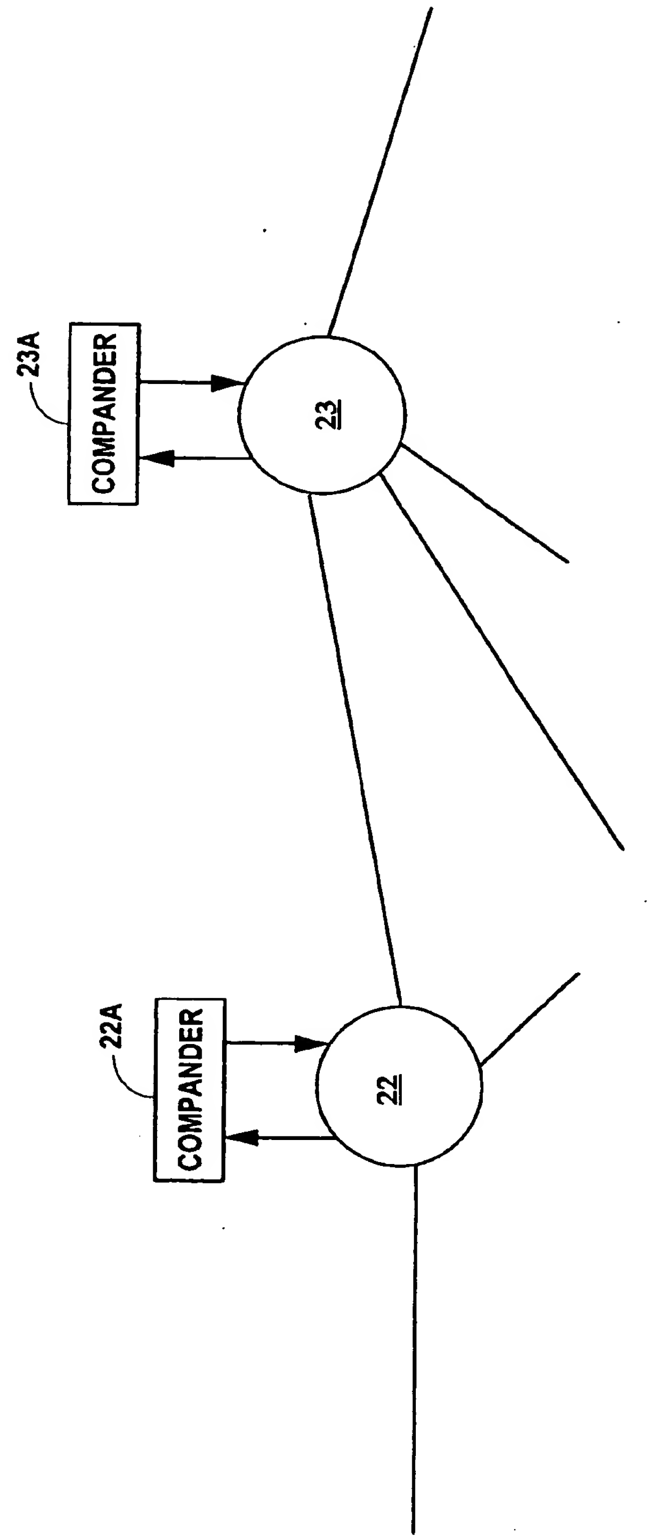


Fig. 7

INTERNET PROTOCOL (IP) TELECOMMUNICATION

This application claims benefit of Provisional Application No. 60/091457 filed Jul. 1, 1998.

The invention generally relates to long distance communications and more particularly to apparatus and methods for assessing whether or not a link or a path between endpoints in an internet protocol (IP) network is potentially suitable for transmission of telecommunications signals of synchronous signals origin. Examples of sources and destinations of such signals may include, but are not limited to, any of voice, data and image terminal apparatus or any combination thereof.

BACKGROUND

In telecommunications, time is an essential component in the information content of signals representing audible sounds and in the signal formats of many visibly reproducible signals, such as television signals. By contrast in data communications, preservation of the time component is not nearly so important. The field of telecommunications has long been operated on the basis of circuit switching principles for providing voice communications. In the later half of this century circuit switching networks have carried an ever increasing volume of data communications. The typical telecommunications digital network for communications of voice, digitally encoded in accordance with a pulse code modulation (PCM) standard, provides a continuous bit rate service that is concatenated as nx64 Kb/s channels. Such telecommunications facilities and networks are said to be circuit switched or synchronous networks, which by their physical natures are most suited to transporting signals between communications terminals which produce synchronous signals, for example telephone speech signals. The primary characteristic of circuit switching is that when one or more physical channels are assigned to a given communication circuit, to provide a service, that channel assignment is reserved for the exclusive use of that service continuously throughout the duration of the service provision. This characteristic of circuit switching is substantially irrelevant for data communications and is of such cost that alternatives, known as packet switched networks, have been developed for the express purpose of providing less costly data communications.

Some time ago, a packet switch, with the trademark SL1 was introduced by the assignee, for improving the efficient transport of data signals. In contrast to the steady repetitive nature of PCM signals, data signals for the most part, are bursty or asynchronous in nature. Thus to accommodate the efficient transmission of data signals, a data burst is arranged into a packet of convenient length along with a header which specifies a destination. After a packet has been assembled, a high speed transmission path is allocated, only for a time sufficient to transport the packet of data toward its destination. During the packet transport, the packet is in sole possession of the transmission path. After the packet is transported, the transmission path is available for the transport of another packet, possibly from a different source. The event of transporting at least one packet of data from a point of origin to a point of destination is termed a data call, however, the number of data packets transmitted throughout the duration of a data call is generally unlimited. Packet networks operated in accordance with the internet protocol (IP) have recently become the data communications equivalent of the publicly accessible switched telephone networks (PSTNs). Users of data communications services commonly refer to the "internet" in the same fashion as users of voice communications refer to the "telephone". If considered on a basis of bulk data transport per unit of cost, the transport of information signals using the IP is very economical as compared with the PSTN. Although digitized voice can be transported via a packet network operated in accordance with the IP, the wide variances of delay caused by the operating characteristics of an IP network tend to deteriorate, distort and occasionally even obliterate the time component. Interrupted, delayed and out of sequenced reception of voice signal packets are common occurrences from time to time in a typical packet system, particularly during higher traffic periods. In other words the IP does not provide a consistent quality of service (QOS) for voice communications and the like.

The general evolution of packet systems toward functionality as broad band carriers of information of synchronous origin is exemplified in a paper by A. Thomas et al, titled Asynchronous Time-Division Techniques: An Experimental Packet Network Integrating Video communication, which was published at the 1984 International Switching Symposium, May 7–11 in Florence Italy. Another example was published in a 1987 IEEE paper by Jean-Pierre Coudreuse and Michel Serval, titled Prelude: An Asynchronous Time-division Switched Network.

More recently, a broadband communications standard for supporting a variety of both synchronous and asynchronous communication requirements has been widely adopted by telecommunications providers, and is now referred to as the asynchronous transfer mode (ATM) of telecommunications. The recommended standards are defined by the ATM Forum and are available from several publishers including Prentice Hall of Englwood Cliffs, N.J. 07632, under the title ATM User-Network Interface Specification Version 3.0 (ISBN 0-13-2258633). One commercially available product is sold by the assignee with the trademark Magellan. Networks operable in the ATM standard are usually termed ATM systems or ATM networks. ATM systems are sophistically compromised to preserve the essence of the time component in synchronous signals yet to some extent reap the economies of packet switching. However, as the IP is strictly directed to the efficient transport of data through any packet network facility, accordingly the IP does not take advantage of the ATM potential for preserving the time component.

The quality of audible speech, reproduced from IP transport, may approach the quality of signals transmitted via the PSTN. Such is usually contingent upon all of the packet network facilities, involved with the transport of the signals, being operated at small fractions,of their capacities. Otherwise the quality may degenerate such that verbal information becomes unintelligible. Never the less, it has become commonplace for some personal computer users to link with an IP network for telephone like voice communications, as well as for data communications. Economies envisaged with utilization of IP networks for synchronous signals communications, have generated considerable development in adaptations of end terminal facilities and software. These adaptations provide degrees of compensation for the irregular delays in packet transport to improve the quality of audibly reproduced speech.

Commercial entities which depend heavily upon telecommunications usage, for their activities, spend significantly upon purchases of telephony services from PSTN and other circuit switched telephone service providers. For some time they have considereded the IP, wishing it were a practical alternative. Improvements in the adaptation of end terminal facilities and software for telephone conversations have made the IP network a potentially practical alternative to the

PSTNs. Users depending heavily upon telecommunications find the potential low cost of usage of IP in comparison with the PSTN to be very attractive. Nevertheless, this attraction is tempered with the recognition that from time to time one or more links between the network endpoints related to a telephone call may provide such poor QOS as to be unacceptable. Furthermore the QOS can be unpredictably variable, changing from good to marginal to bad and to good again within an hour.

The effects of packet delays and losses as well as end terminal clocking dissimilarities are discussed in a previous U.S. patent application Ser. No. 08,982,925 assigned to Northern Telecom Ltd., the assignee of this application. The previous application teaches improvements useful in telephone facilities, terminals and personal computers which reduce the potentially deleterious effects of an IP network involved in speech transmission. However, there are practical limits to the effectiveness of these improvements, while there is no limit to the degradation of the performance of an IP link in a packet network.

In the jargon of the IP, a real-time transport protocol has been introduced to distinguish signals of synchronous origin from typical data signals. A signal of synchronous origin is usually referred to as a real-time transport protocol (RTP) stream. This does nothing to expedite the regular transport of these signals, however it has permitted the use of a real-time transport control protocol (RTCP). The RTCP is one of several protocols useful for collecting data relative to characteristics having been inflicted upon RTP streams, while traversing the IP network. Both the RTP and the RTCP are published in standards recommended by the International Telecommunications Union. There are also software tools available which will analyze the collected data and interpret the characteristics of data collected by the RTCP. In other words, such software tool assesses the QOS being momentarily provided via a path of propagation through the IP network. Examples of these software tools are available under the trademarks of V/IP Trunk from Micom, and IP Telecommuter and Road Warrior both from Northern Telecom. Each of these software mechanisms includes functions which take a measure of the performance of an IP connection for synchronous data, such as a telephone call after the call has progressed to a conversation. Of course if the connection provides transport of inadequate quality for real-time voice, the parties to the conversation may very well realize it without the benefit of what is essentially a post performance assessment of the RTP stream. At least one party will likely notice the conversation received from the other party as being delayed, broken or otherwise unsatisfactory. Consequently, as far as telephone voice communications are concerned, many if not most commercial entities are reluctant to becoming committed internet telephony users, as for purposes of their activities, a guaranteed QOS similar to the QOS of the PSTN is virtually essential.

SUMMARY OF THE INVENTION

In accordance with the invention, an assessment of a probable QOS for routing an intended telephony call via an IP network is acquired, apriori to actually establishing the telephone call.

In one example, RTCP information, about RTP streams having recently traversed links related to a specific path in the IP network, is gathered to determine, apriori setup of a requested call, a QOS of an IP network path.

More particularly, a quality of service (QOS) server, in combination with an IP network, is responsive to IP path definitions provided thereto from a telecommunications entity coupled with the IP network, for from time to time collecting data relative to characteristics of real-time transport protocol (RTP) streams used for transporting real-time audio data via one of more links in an IP network path.

In one application of the invention a call centre including agent stations, is responsive to data relative to characteristics of real-time transport protocol streams used for transporting real-time audio data via IP paths in an IP network, for selectively postponing an IP telephone call setup with one of the agent stations in an instance wherein the data indicates a probable QOS of less than a predetermined QOS.

In one example the call centre is operative in combination with a quality of service (QOS) server to accommodate outgoing call completions via either of a PSTN and a packet switched network. The call centre comprises a telephone circuit switching network being coupled via trunk circuits to a PSTN, and being coupled via a gateway means to a packet switched network, the telephone circuit switching network being operable to provide communications channels between any of a plurality of agent stations and the gateway means, and to provide communications channels between any of the plurality of agent stations and the trunk circuits. A call controller directs the operations of the telephone circuit switching network for setting up and tearing down telephone calls, such that calls for completion via the trunk circuits are preceded by signalling information specifying at least the telephone number of a called party, and such that calls for completion via the gateway means are preceded by signalling information as to the telephone number of a called party as well as a corresponding IP network endpoint address. An automatic dialler is coupled to communicate with the call controller for providing telephone numbers in association with endpoint addresses when requested by the call controller, and is coupled with the QOS server. The automatic dialler comprises a number list for storing a plurality of telephone numbers along with corresponding endpoint addresses of parties to be called. A calling list controller reads and writes the number list, and apriori to providing a telephone number for use by the call controller, the calling list controller is operable to request and receive QOS information from the QOS server in relation to the endpoint addresses of the call centre's gateway and an endpoint associated with said telephone number. The calling list controller is further operable to decline provisioning of said telephone number in consequence of a value of said QOS information. Hence in an IP voice call, a predetermined acceptable QOS can be virtually guaranteed.

The automatic dialler in one example includes a deferred list and a called list, in addition to the number list. The deferred list, is for recording telephone numbers for which provisioning was declined. The called list is for storing each telephone number read in the telephone number list and resulting in an IP telephone call setup with a called party having been performed. In this example the calling list controller is responsive to a completion of a reading of all the numbers in the telephone number list for functionally substituting the deferred list; and is responsive to a signal from an agent station during a progress of a telephone connection following the call setup for causing the telephone number of the called party to be recorded in the deferred list.

A method for routing a telephone call via an IP network, is performed in response to a calling party having initiated a call to a called party. The method includes the step of determining at least one IP network link for transport of packets from the calling party to the called party, and at least one IP network link for transport of packets from the called

party to the calling party. In accordance with the invention the method comprising the further steps of:

collecting data relative to characteristics of any real-time transport protocol (RTP) streams having recently been transported via said links;

generating a historical quality of service (QOS) value from the collected data; and

setting up the call between the calling and called parties via the IP network, contingent upon the historical QOS value being of at least a predefined QOS;

whereby the setting up an IP network telephony call with a poor QOS is substantially avoided.

The invention provides a method for estimating a telephony quality of service between endpoints in an internet protocol (IP) network. The method requires a data storage facility coupled with the IP network to act as a quality of service (QOS) server. The method is responsive to an exchange of signal streams of synchronous origins at an endpoint and comprises the steps of:

a) gathering performance information for a signal stream received at the end point,

b) gathering route information identifying at least one route having been traversed by said signal stream,

c) transferring the gathered information to the data storage facility; and

in response to each transfer of said gathered information, at the data storage facility,

d) storing the performance information, and the route information;

whereby the stored information are available apriori a call setup for a request of telephone service involving endpoints having at least a potential transport route in common with a route identified in the stored route information.

Following the request for telephone service, the method comprises the further steps of:

e) identifying IP network endpoints of the calling and called parties,

f) transmitting a request for QOS information relative to each of the identified endpoints,

g) responsive to said request, in the QOS server, reading any QOS information for which there is at least a potential transport route in common with a route identified in the stored route information, and

h) transporting any QOS information, read in step d), to the requester,

whereby set up of the requested telephone call via the IP network may be declined if the QOS information appears to indicate less than a prescribed QOS.

INTRODUCTION OF THE DRAWINGS

Example embodiments of the invention are discussed with reference to the accompanying drawings in which:

FIG. 1 is a block schematic diagram of an IP network and a PSTN network for coupling telecommunications devices and facilities such that telecommunications may be conducted between any of the telecommunications devices and facilities via either of the IP network and the PSTN, in accordance with the invention:

FIG. 2 is a block schematic diagram showing more detail as to a private branch exchange (PBX) used in FIG. 1;

FIGS. 3 and 4 are flow diagrams which illustrate a method by which a PBX call centre feature utilizes the IP network of FIG. 1 for telecommunications, consequent upon QOS information;

FIG. 5 is a flow diagram which illustrates a method for amassing QOS information in relation to endpoints and routes in the IP network in FIG. 1;

FIG. 6 is a flow diagram which illustrates a method by which a telephone call is setup, consequent upon QOS information in relation to endpoints and routes in the IP network in FIG. 1; and

FIG. 7 is a block schematic diagram which illustrates an example of a network resource useful in combination with the IP network of FIG. 1.

DESCRIPTION OF THE EXAMPLE EMBODIMENT

Points of origin of synchronous telecommunications signals, potentially transmittable via the IP network, are located at scattered locations around the globe and may transmit at any time to one or more points of destination. These points of origin and destination may include telecommunications devices or telecommunications facilities, as well as computers ranging from small personal computers to large main frame computers. In FIG. 1 a public switched telephone network (PSTN) 10 is shown with a multitude of telephone devices 9, connected thereto via a corresponding multitude of telephone lines 8. Some examples of telephone devices include but are not limited to subscriber telephones, facsimile machines, image cameras and displays, modem interfaced apparatus and the like. The PSTN 10 is also illustrated as connected to provide telephone service for a PBX 14 via a trunk facility 14*b* and a key telephone system (KTS) 15. It can be summarized that the typical PSTN can provide telecommunications services for a wide variety of communications facilities and devices not all of which are shown in the figures.

Access between the PSTN 10 and an IP network 20 is typically provided at multiple locations across a continent and is exemplified in FIG. 1 as a communications link 12 terminated at a gateway 24. The gateway 24 transfers information between the operating signal formats of the PSTN and the IP network, the former typically being a TDM PCM format, and the later being a data packet format in accordance with the IP. The PBX 14 is coupled to a router 24 in the IP network 20 via a link 14*a* and is also coupled to the PSTN 10 via a trunk 14*b*. A QOS server 18 is coupled via a link 18*a* with an IP network router 21, to be generally available as a network resource. Another QOS server 19 is dedicated to serving the PBX 14. The dedicated QOS server 19 is connected to the PBX 14 via a link 14*c* and coupled to an IP network router 29 via a link 19*a*. By way of example, a network 13 of personal computers is shown linked via a link 13*a* to an IP network router 28, and coupled to the PSTN 10 via a trunk 13*b*. The IP network 20 is also depicted as including paths, routes or links intersecting at IP network nodes 22, 23, 26, and 27. In actual fact packet networks have multitudes of gateways or service endpoints and a multitude of nodes linked therebetween, however this is not further discussed, as the particular structures representative of IP operative networks are not pertinent to an appreciation of the invention.

Both the PSTN 10 and the IP network 20 are capable of providing communications between a multitude of entities which may be coupled thereto, however, as discussed in the background, the IP network operates by data packet transmissions while the PSTN operates by switched circuit synchronous signal transmissions.

As before discussed the economies of using the IP network for communications has lead to various arrangements

for minimizing the deleterious intrusive nature of the IP network in regards to the transmission of voice signals and the like in a so-called RTP stream. Also as before mentioned there are also software tools available which will analyze the collected data to determine the characteristics of an RTP stream. In this example the QOS servers **18** and **19** are each implemented by computer apparatus (not shown) which is dedicated to collecting information about the flow of any RTP streams in the IP network **20**. The QOS server **18** is available to answer requests from other entities connected to the IP network **20** by transmitting the latest data collected about the flow of RTP streams via any specified link or between specified endpoints. The requesting terminal or device may then determine the probable QOS of the link or links in the IP network path over which the signals will be transported during a telephone call. The KTS in this case may conveniently be a Norstar, available from Northern Telecom Ltd. The KTS **15** uses the QOS information to decide if a requested telephone call between the endpoints of origin and destination is viable. It is envisaged that any end point devise that may be served by an IP network connection and involved with the origination of synchronous signals will include means by which data representative of recent link performance received from a common resource IP network QOS server is analyzed so that a calling party can be come aware of the probable QOS of the moment and choose either the IP network or the PSTN for completing the call.

The dedicated QOS server **19** is exclusively available to answer requests from the PBX **14**, in a similar manner, by transmitting the latest data collected about the recent flow of any RTP streams having been transported by a link or group of links that might be used to complete a telephone call, originating from the PBX **14**. Using an appropriate software tool, for example IP Telecommute, the PBX **14** determines the probable QOS of the link or links in the IP network path over which the signals will be transported during the telephone call. The QOS information may be relied upon to determine the disposition of the initiated telephone call, for example to be to be one of routing through the IP network **20**, routing through the PSTN **10**, or simply refusing to complete the call.

Referring to FIG. **2**, the PBX **14** introduced in FIG. **1**, is configured in a typical manner to provide a call centre which includes a PBX switching network **31** operated under the direction of a PBX call controller **33**. For example in a telemarketing function any of a plurality of agent stations **32** may be coupled to the PSTN **10** via one of trunk circuits **34** and a channel in the trunk **14*b***, or alternately to the IP network **20** via an Ethernet gateway **35** and the link **14*a***. The PBX call controller **33** is responsive to call requests, for calling distant parties, as defined by telephone numbers supplied via a command status link **36** by an automatic dialler **40**. When a party answers the call is quickly completed to an idle attendant at an agent station. The PBX call controller **33** is capable of processing calls at a rapid rate. However it is normally programmed to pace the call centre such that an answering called party will have the impression that the agent had dialed and was waiting for the party to answer, when in fact the agent may have disconnected from the conversation of a previous call only a moment before.

In this particular example, the automatic dialler **40** is provided as a server element connected to the call controller **33** via a command status link **36**. The PBX in this case may conveniently be a Meridian 1, available from Northern Telecom. The automatic dialler **40** includes a calling list controller **41** which manipulates a data base. The data base would normally be contained in a single random access memory device. For ease of illustration, the data base is depicted as being three separate memory devices, each of which is functionally labeled as a number list **45**, a called list **46**, and a deferred list **47**, each being coupled to the calling list controller **41** via a data bus **42** and a control bus **43**. It is the calling list controller'responsibility to present numbers for calling one after another to the PBX call controller **33** via the command status link **36**.

Operation of the PBX is discussed in more detail with reference to FIGS. **3** and **4**. A memory element in the automatic dialler **40** is segmented into the number list **45**, the called list **46**, and the deferred list **47** as shown in function box **51**. To prepare for operation, the function box **52** requires the number list **45** to be loaded with the telephone numbers. The telephone numbers will be those of a selected portion of a population with whom the commercial entity desires to do business or otherwise communicate with. As it is desirous that the IP network **20** be utilized whenever possible, routing information as to appropriate IP network endpoints or gateways is associated with each of the loaded telephone numbers or alternately the telephone numbers are arranged in blocks having common end point addresses. Loading of the telephone numbers and associated endpoint addresses may be performed by physical insertion of a prepared data base in the form of a cassette data tape (not shown) into either of the call controller or the calling list controller. Alternately, such data base information may be received from a remote source (not shown) via the IP network **20**, the Ethernet gateway **35** and thence traverse a TDM signalling and supervision channel in the switching network **31** to reach the call controller **33**.

Preparatory to the call centre function, as shown at box **53**, the calling list controller **41** reads a number and routing information in the number list **45**. The calling list controller **41** compares the routing information with any recent entries in the deferred list **47**, as required in a decision box **54**, to determine if a recently read number with similar routing was declined. If not, according to function box **55**, the routing information is transferred to the QOS server **19**. As required in decision box **56**, if RTP stream data information corresponding to the routing information is of recent record, the QOS server **19** makes the data known to the calling list controller **41**, as shown in box **58**. However if such data is absent or more than say 5 minutes old, for example, the QOS server collects current data related to the endpoints by initiating an artificial session, which mimicks a call involving RTP streams being exchanged between endpoints of the route through the IP network **20**, as shown at **57**. The artificial session data is used to update the QOS server and is communicated to the calling list controller **41**. An alternate result in the decision box **54**, occurs if a recently read number with similar routing was declined and results in a YES. A yes result causes the sequence to transfer at "B" to FIG. **4** function box **65**, which is discussed later.

FIG. **4** is entered via "A" from function box **58** where in accordance with decision block **59**, the calling list controller **41** determines if the probable QOS is of at least a prescribed criterion. If the QOS is satisfactory the calling list controller **41** provides the telephone number and the routing information via the command/status link **36** for use by the PBX call controller **33**, as shown in a function box **60**. In accordance with decision block **61**, after the telephone number and the routing information is confirmed as having been accepted by the call controller **33**, an other number and endpoint information are read from the number list **45** as required in function box **64**, and the sequence transfers via "C" to FIG.

3. At the same time the list controller 41 waits to receive an indication from the call controller 33 that the call specified by the previously read telephone number has resulted in a telephone call having progressed to being answered. If and when such occurs, as indicated in function block 62, the corresponding telephone number and endpoint information are written in the called list 46. As an option, after the conversational portion of the call, before going ON HOOK the agent may signal a recall request from the station. The call controller 33 signals the list controller 41 to write the telephone number and the associated endpoint information in a recall list (not shown) for a second call setup request at some later time.

The function box 65, is responsive to a NO assertion at the function block 59, or to a YES assertion as the function. block 54 to require the telephone number and the associated endpoint information be written in the deferred list 47. Decision block 66 tests to see if all the numbers in the number list have been read. If YES, then the calling list controller checks to see if the number list has already been substituted by the deferred list 47, as specified in decision block 67. If NO then the deferred list is substituted and the sequence of functions transfers to function box 64 as required by function box 68. In other words, when the number list 45 has been read in its entirety, the calling list controller 41 substitutes the deferred list 47 instead of the number list 45 as the source of numbers for calling. Transfers of numbers to the PBX call controller 33 are treated in like manner as before described. On the other hand if YES, then the reading of the deferred list 47 has been completed and the process ends at 69.

Thereafter numbers from the call again list may be used to drive the call centre or the number list 45 can be loaded with a new sequence of numbers and the call centre resumes operation.

In relation to FIGS. 3 and 4, operation of a call centre using the IP network for telecommunications has been discussed. In particular telephone numbers are specified for call setup using the IP, only after a probably adequate QOS has been determined.

In another embodiment, later discussed with reference to FIG. 7, the QOS server is able to specify an IP network node compander function such that the bandwidth of the RTP stream may be compressed for transport via one or more moderately traffic congested links and thereby improve the perceived telephony service.

One method by which a QOS server amasses IP telephony QOS information is discussed with reference to FIG. 5. QOS information is amassed in a QOS server by monitoring ongoing traffic so that a short history of performances, each related to a specific route, is obtained. Starting at 70, in a normal IP telephone call, RTP signal streams are exchanged at the endpoints associated with the calling and called parties. The endpoints are instructed to utilize the RTCP to generate information as to the characteristics of the streams, and provide this with related path or route information in one or more data packets addressed for the QOS server, as indicated in function boxes 71, 72 and 73. The QOS server stores the information association with the identities of the endpoints of origin as indicated in function box 74. The process ends as indicated at function box 75, but recommences with each initiation of a normal synchronous session, as shown at 70. The information is selectively available for determining a probable QOS for a requested IP telephone call apriori the actual call set up. However, in the event that no traffic of synchronous signals origin has recently occurred over a particular route, the probability of the previously gathered information being somewhat accurate is depreciated. It is therefore considered to be of no practical value and apriori a call setup an attempt to acquire more up to the moment data can be initiated by requesting an artificial synchronous session, as indicated at 70A in FIG. 5. The artificial synchronous session is controlled by the QOS server which requests the endpoints to exchange media pseudo synchronous signal streams, as indicated at function box 71A. The resulting RTP streams must be of sufficient bulk to facilitate a meaningful examination by the RTCP. After the artificial synchronous session has generated some performance information, the session ends, as shown at 72A, and the information is gathered and forwarded to thee QOS server, where it is stored in accordance with the functions illustrated at function boxes 73A, 74A, 73 and 74.

A method by which an IP telephone call is setup, consequent upon the gathered QOS information, is discussed with reference to FIG. 6. In an event of a calling party having dialed or otherwise indicated the telephone number of a called party, as indicated at 80, network endpoints related to the calling and called parties must be identified, as indicated in function box 81. Thereafter QOS information relating to the endpoints is requested from the QOS server, as shown at function box 82. If for example, referring to FIG. 1, the request originates with any of the KTS 15 or telephone line connected devices 9, the request will have been initially transmitted via a TDM channel provided by a local exchange (CO), not shown, in the PSTN 10 to the gateway 24. At the gateway 24 the TDM signal is packetized for transport to the QOS server 18. If for example the request originates at a personal computer in the network of personal computers 13, the request could traverse the PSTN 10 but more efficiently the request is likely to be transported by the router 28 and on via the IP network 20 to the QOS server 18. Referring again to FIG. 6, the QOS server responds to the request by first determining if there is information relating to each of the endpoints within the QOS server, as indicated at a decision block 83. If YES, the QOS information is read, at 84, and examined to see if it is recent, as required at decision block 86. If NO, the artificial synchronous session is initiated as required in a function box 89. If YES, the information is transported toward the requester, as indicated in function box 90. The information is then available for the requesting entity for either indicating the potential QOS, to a user, or available for use directly to automatically determine the progress or disposal of the dialed telephone call.

Again referring to FIG. 6 at the decision block 83, if it is determined there is no information in relation to each of the endpoints, a further decision is specified in a block 87. In this block it is determined if QOS information relating to the endpoints might be inferred. Inference is attractive as it is much faster and uses less resource, compared to initiating an artificial synchronous session. Inference is possible if the endpoint locations are considered, in order to determine a likely route therebetween. If there is QOS recent information for the likely route, it is inferred to be appropriate QOS information at function box 88 and is forwarded to the requester. If the QOS is not inferable an artificial synchronous session is initiated as required at the function box 89.

Referring to FIG. 7, the IP network nodes 22 and 23 as illustrated in FIG. 1, are also connected in combination with companders 22A and 23A respectively. Companding is a well known function by which the bulk of an information signal is compressed for transmission and subsequently expanded, complimentary to the compression function, for reception and use while maintaining virtually all of the

information content. In this example the compander is a device which performs either of a selected compression function or a complimentary expansion function, upon RTP packets of a particular IP telephone call. If a QOS for a requested IP telephone call appears to be somewhat less than a preferred QOS in the link between the nodes 22 and 23, while any alternative links are unacceptable, companding is-used to reduce the signal bulk for transport. Compression of the RTP stream reduces the amount of data signal for transporting the information. Hence the time duration for transport of RTP packets is reduced and accordingly the exposure of the information to IP degradation while traversing the link is reduced. In the example of FIG. 2, the network companding resource is used if the calling list controller **41** receives QOS information which indicates the historical QOS value of a link to be less than the preferred QOS value, but greater than a lesser predefined QOS value. The calling list controller **41** provides link routing instruction along with the endpoint addresses and the telephone number for use by the call controller **33**. The link routing instruction includes a request for allocating the network companding resource to the RTP streams to be transported via the link for the duration of the telephone call. By this means parties to the IP telephone call will experience a better QOS than would be available without the companding resource.

In the forgoing discussion, examples of a QOS server, in association with a packet switched network operated with the IP for apriori determining a probable QOS for a requested IP telephone call, have been illustrated. It is envisaged that in the light of this discussion QOS servers will be provided as a common resource in IP networks to better facilitate IP telephony. The principles discussed herein extend generally to signals of synchronous origin, be these conveyers of whatever information. Examples of sources and destinations of such information may include, but are not limited to any of, voice, data, and image terminal apparatus, as well as commonly accessible network processing and companding resources. It is also envisaged that future telephone apparatus and other terminal devices will include apparatus and software in accordance with the previously mentioned patent application Ser. No. 08,982, 925. In addition thereto such telephone apparatus and terminal devices will include means for utilizing IP telephony QOS information. With the knowledge of this discloser those persons skilled in the field of IP telephony will realize other variations and embodiments within the spirit and scope of the invention.

What is claimed is:

1. A method for routing a telephone call via an IP network, responsive to a calling party having initiated a call to a called party, wherein the method includes the step of determining at least one IP network link for transport of packets from the calling party to the called party and at least one IP network link for transport of packets from the called party to the calling party, the method comprising the further steps of:

collecting data relative to characteristics of any real-time transport protocol (RTP) streams having recently been transported via said links;

generating a historical quality of service (QOS) value from the collected data; and

setting up the call between the calling and called parties via the IP network, contingent upon the historical QOS value being of at least a predefined QOS;

whereby the setting up an IP network telephony call with a poor QOS is substantially avoided.

2. A method as defined in claim **1** wherein said predefined QOS is a first predefined QOS value, and in an event where the historical QOS value of any link being less than the first predefined QOS value, but greater than a lesser predefined QOS value, allocating a network companding resource to the RTP streams to be transported via said link; and setting up the call between the calling and called parties via said links.

3. A quality of service (QOS) server, in combination with an IP network, being responsive to IP path definitions provided thereto from a private branch exchange (PBX) coupled with the IP network, for from time to time collecting data relative to characteristics of real-time transport protocol streams used for transporting real-time audio data via one or more links in an IP path, wherein said PBX includes an automatic dialler and wherein said IP path definitions are provided by the automatic dialler in the form of IP network endpoint addresses, the QOS server providing collected data relative to said endpoints whereby an automatic dialler defined call set up via the IP network is declined if the QOS is unsatisfactory.

4. A call centre including agent stations, the call centre being responsive to data relative to characteristics of real-time transport protocol streams used for transporting real-time audio data via IP paths in an IP network, for selectively postponing an IP telephone call setup with one of the agent stations in an instance wherein the data indicates a probable QOS of less than a predetermined QOS.

5. A call centre as defined in claim **4** comprising:

a telephone number list for storing prescribed telephone numbers of parties to be called, whereby the call centre performs a call setup in response to a telephone number having been read in the telephone number list;

a deferred list, for recording said telephone number in the event of the call setup being postponed; and

a called list for storing each telephone number having been read with the result of an IP telephone call setup with a called party having been performed.

6. A call centre as defined in claim **5** comprising:

means responsive to a completion of a reading of all the numbers in the telephone number list for functionally substituting the deferred list.

7. A call centre as defined in claim **6** comprising:

means being responsive to a signal from an agent station during a progress of a telephone connection following the call setup for causing the telephone number of the called party to be recorded.

8. A call centre being operative in combination with a quality of service (QOS) server, to accommodate outgoing call completions via either of a PSTN or a packet switched network, comprising;

a telephone circuit switching network being coupled via trunk circuits to the PSTN, and being coupled via a gateway means to the packet switched network, the telephone circuit switching network being operable to provide communications channels between any of a plurality of agent stations and the gateway means, and to provide communications channels between any of the plurality of agent stations and the trunk circuits;

a call controller for directing the operations of the telephone circuit switching network for setting up and tearing down telephone calls, such that calls for completion via the trunk circuits are preceded by signalling information specifying at least the telephone number of a called party, and such that calls for completion via the gateway means are preceded by signalling information as to the telephone number of a called party as well as corresponding IP network endpoint address;

an automatic dialler coupled to communicate with the call controller for providing telephone numbers in association with endpoint addresses to the call controller, and coupled with the QOS server, the automatic dialler comprising:

a number list for storing a plurality of telephone numbers along with corresponding endpoint addresses of parties to be called;

a calling list controller for reading and writing the number list and apriori to providing a telephone number for use by the call controller, the calling list controller being operable to request and receive QOS information from the QOS server in relation to the endpoint addresses of said gateway and an endpoint associated with said telephone number, and being operable to decline provisioning of said telephone number in consequence of said QOS information suggesting an insufficient QOS.

9. A call centre as defined in claim 8, the automatic dialler further comprising;

a deferred list, for recording said declined telephone number;

a called list for storing each telephone number read in the telephone number list and resulting in an IP telephone call setup with a called party having been performed; and the calling list controller further comprising:

means responsive to a completion of a reading of all the numbers in the number list for functionally substituting the deferred list; and

means being responsive to a signal from an agent station during a progress of a telephone connection following the call setup for causing the telephone number of the called party to be recorded.

10. A method for estimating a telephony quality of service (QOS) between endpoints in an internet protocol (IP) network, comprising the steps of:

coupling a data storage facility with the IP network to act as a quality of service (QOS) server; and

responsive to an exchange of signal streams of synchronous orgins at an endpoint,

a) gathering performance information for a receive signal stream,

b) gathering route information identifying at least one route having been traversed by said signal stream,

c) transferring the gathered information to the data storage facility; and in response to each transfer of said gathered information, at the QOS server,

d) storing the performance information, and the route information;

whereby the stored information are available apriori a call setup for a request of telephone service involving endpoints having at least a potential transport route in common with a route identified in the stored route information, said performance information and route information being operative to determine whether a QOS associated with said telephone service would be greater or less than a prescribed QOS.

11. A method as defined in claim 10 wherein following said request for telephone service, the method comprising the further steps of:

e) identifying IP network endpoints of the calling and called parties,

f) transmitting a request for QOS information relative to each of the identified endpoints,

g) responsive to said request, in the QOS server, reading any QOS information for which there is at least a potential transport route in common with a route identified in the stored route information, and

h) transporting any QOS information, read in step d), to the requester,

whereby set up of the requested telephone call via the IP network may be declined if the QOS information appears to indicate less than a prescribed QOS.

12. A method as defined in claim 10, comprising the further step of:

j) coincident with step d), storing the time of each occurrence of step d) in association with said information.

13. A method as defined in claim 12, where in an event of a request of telephone service involving endpoints having at least a potential transport route in common with a route identified in the stored route information, the method comprising the further steps of:

k) dependent upon the time at which the information was stored in step d) not being recent enough, initiating an artificial synchronous session wherein signal streams, simulated to be similar to signal streams of synchronous origins, are exchanged between the endpoints,

m) gathering performance information related to the signal streams received at the endpoints,

n) gathering route information for the signal streams received at the endpoints, and

performing the steps of;

c) transferring the gathered information to the data storage facility; and

in response to each transfer of said gathered information, at the QOS server,

d) storing the performance information, and the route information.

14. A method as defined in claim 11 wherein step g) is not performed as there is no potential transport route in common with a route identified in the stored route information, the method comprising the further step of:

initiating an artificial synchronous session causing exchanges of signal streams, simulated to be similar to signal streams of synchronous origins, between the endpoints, and thereafter, performing steps beginning with step m) defined in claim 13.

15. A method as defined in claim 10 wherein following said request for telephone service the method comprising the further steps of:

e) identifying IP network endpoints of the calling and called parties,

f) transmitting a request for QOS information relative to each of the identified endpoints,

p) in the QOS server being responsive to said request such that if there is QOS information for said endpoints transporting the QOS information to the requester.

16. A method as defined in claim 15 where in an event there is no information for one of the endpoints, determining if there is a likely route between the endpoints for which there is QOS information, if so and if the QOS information is recent, inferring the QOS information is appropriate and transporting the inferred QOS information to the requester.

17. A method as defined in claim 16 where in an event there is no inferable information for said endpoint, initiating an artificial synchronous session to acquire QOS information for the endpoint, and transporting the acquired QOS information to the requester.

18. A method as defined in claim 15 where in an event there is no information for one of the endpoints, determining if there is QOS information for a route in common with at least one link through which a signal stream destined for the

endpoint would likely be transported, and if so transporting the QOS information to the requester.

19. A method as defined in claim 18 where in an event there is no route in common for said endpoint, initiating an artificial synchronous session to acquire QOS information for the endpoint, and transporting the acquired QOS information to the requester.

20. A method as defined in claim 11, comprising the further step of:

j) coincident with step d), storing the time of each occurrence of step d) in association with said information.

21. A method as defined in claim 20, where in an event of a request of telephone service involving endpoints having at least a potential transport route in common with a route identified in the stored route information, the method comprising the further steps of:

k) dependent upon the time at which the information was stored in step d) not being recent enough, initiating an artificial synchronous session wherein signal streams, simulated to be similar to signal streams of synchronous origins, are exchanged between the endpoints,

m) gathering performance information related to the signal streams received at the endpoints,

n) gathering route information for the signal streams received at the endpoints, and

performing the steps of;

c) transferring the gathered information to the data storage facility; and

in response to each transfer of said gathered information, at the QOS server,

d) storing the performance information, and the route information.

* * * * *

US006591382B1

(12) **United States Patent**
Molloy et al.

(10) **Patent No.: US 6,591,382 B1**
(45) **Date of Patent: Jul. 8, 2003**

(54) **PERFORMANCE IMPROVEMENT OF INTERNET PROTOCOLS OVER WIRELESS CONNECTIONS**

(75) Inventors: **Walter Molloy**, San Diego, CA (US); **Thomas P. Trotta**, Encinitas, CA (US); **Donald B. Eidson**, San Diego, CA (US)

(73) Assignee: **Skyworks Solutions, Inc.**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/375,607**

(22) Filed: **Aug. 17, 1999**

(51) **Int. Cl.**[7] **G06F 11/00**
(52) **U.S. Cl.** **714/704**
(58) **Field of Search** 714/704, 708, 714/774; 370/342, 468; 358/412

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,905,741 A	* 5/1999	Matsukuma et al.	....	714/758
6,038,452 A	* 3/2000	Strawczynski et al.	.....	455/403
6,069,924 A	* 5/2000	Sudo et al.	....	375/330
6,167,039 A	* 12/2000	Karlsson et al.	....	370/320
6,272,190 B1	* 8/2001	Campana, Jr.	....	375/347
6,314,535 B1	* 11/2001	Morris et al.	....	370/468
6,347,081 B1	* 2/2002	Bruhn	....	370/337

OTHER PUBLICATIONS

Badrinath et al., "Handling mobile clients: A case for indirect interaction," *Proceedings of the 4th Workshop on Workstation Operating Systems* pp. 91–97, Napa CA, Oct. 1993.

Balakrishnan et al., "Improving reliable transport and hand-off performance in cellular wireless networks," *Wireless Networks* 1(4):469–481 (1995).

Caceres et al., "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE Journal on Selected Areas in Communication* 13(5):850–857(1995).

Karn, "The Qualcomm CDMA Digital Cellular System," *Proceedings of the USENIX Molile & Location–Independent Computing Symposium*, pp. 35–39, Cambridge MA, Aug. 1993.

Saltzer et al., "End–to–end arguments in system design," *ACM Transactions on Computer System* 2(4):277–288 (1984).

Tanenbaum, "The OSI reference model," *Computer Networks* 3rd Ed. p. 29.

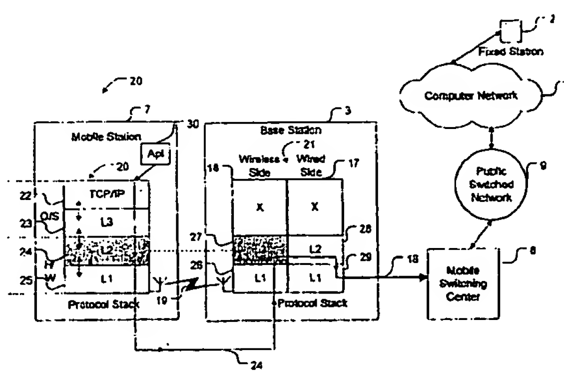
* cited by examiner

Primary Examiner—Albert Decady
Assistant Examiner—Esaw Abraham
(74) *Attorney, Agent, or Firm*—Farjami & Farjami LLP

(57) **ABSTRACT**

A system and method for providing improved performance using TCP/IP protocols over wireless networks that can be implemented entirely within the link layer of a protocol stack. The system and method responds to low signal levels caused by weak and fading wireless connections by maintaining throughput and circumventing inappropriate instances of TCP/IP congestion avoidance mode. At least two selectable service protocols, comprising at least one selectable basic error-detecting/correcting protocol and at least one selectable robust error-detecting/correcting protocol, are implemented within link layers of both the mobile station and the base station. A quality of service monitor installed within the link layer monitors signal quality. When the quality of service monitor detects a signal quality that falls below a predetermined threshold or predicts a future signal degradation, the mobile station switches to the robust error-detecting/correcting service protocol and informs the base station of the changeover. The base station similarly switches to the robust error-detecting/correcting service protocol, beginning at a predetermined data frame. If the quality of service rises above a second predetermined threshold, the basic error-detecting/correcting protocols are restored both within the mobile and base stations.

70 Claims, 12 Drawing Sheets



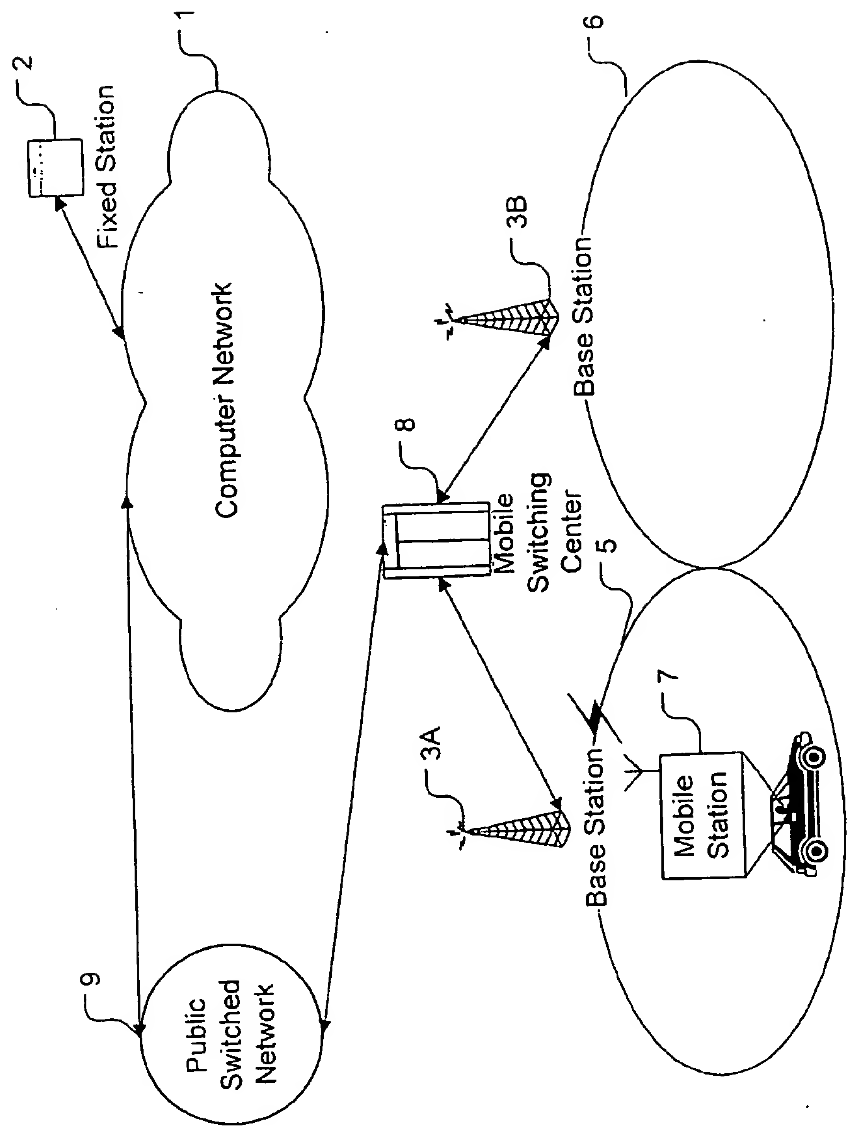

2
Fixed Station
1
Computer Network
9
Public Switched Network
8
Mobile Switching Center
3B
Base Station
6
3A
Base Station
5
7
Mobile Station

Fig. 1

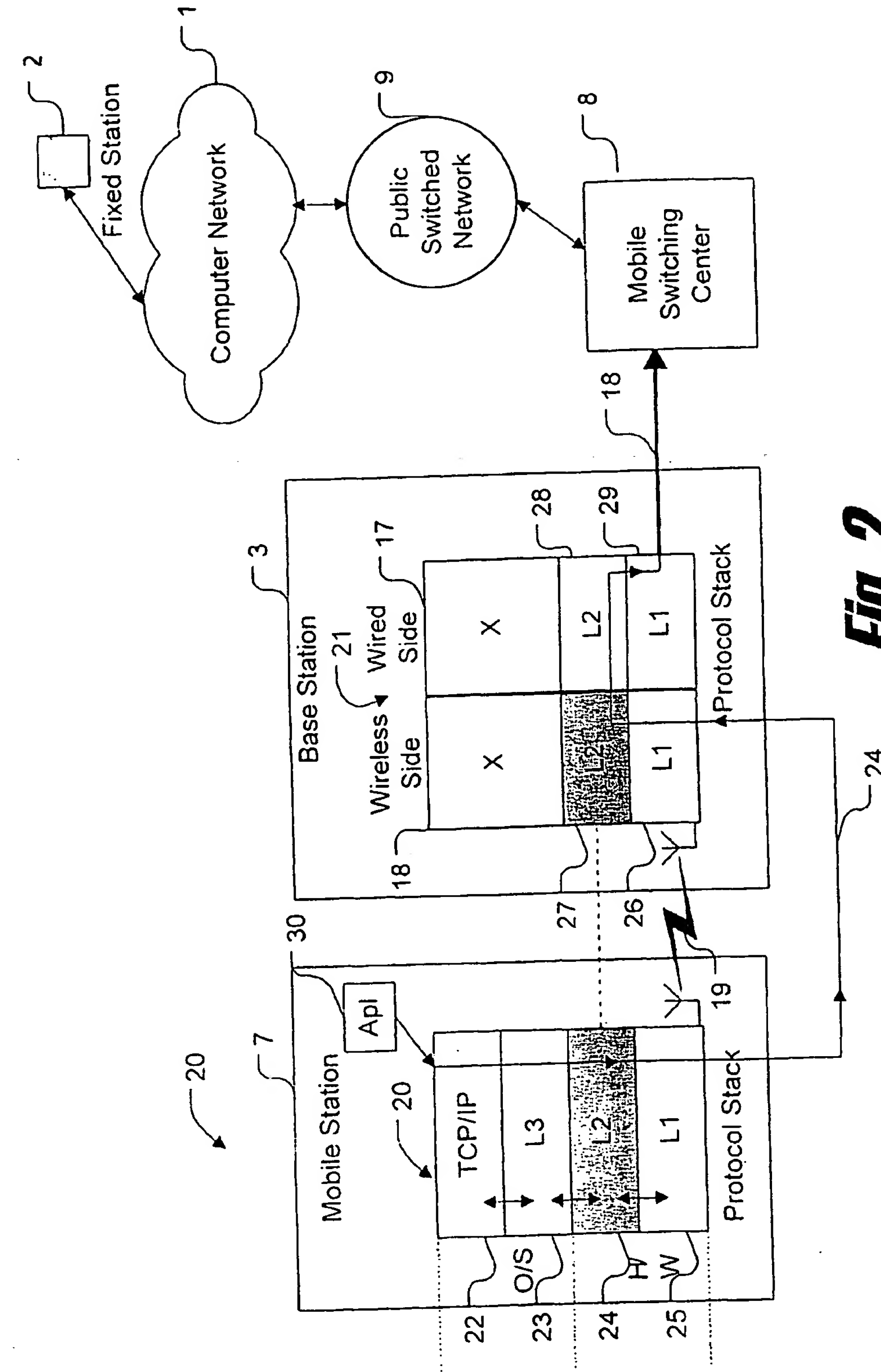

Fixed Station
Computer Network
Public Switched Network
Mobile Switching Center
Base Station
Wireless Side
Wired Side
X
X
L2
L2
L1
L1
Protocol Stack
Mobile Station
Apl
TCP/IP
L3
L2
L1
Protocol Stack
O/S
H/W
1
2
3
7
8
9
17
18
18
19
20
20
21
22
23
24
24
25
26
27
28
29
30

Fig. 2

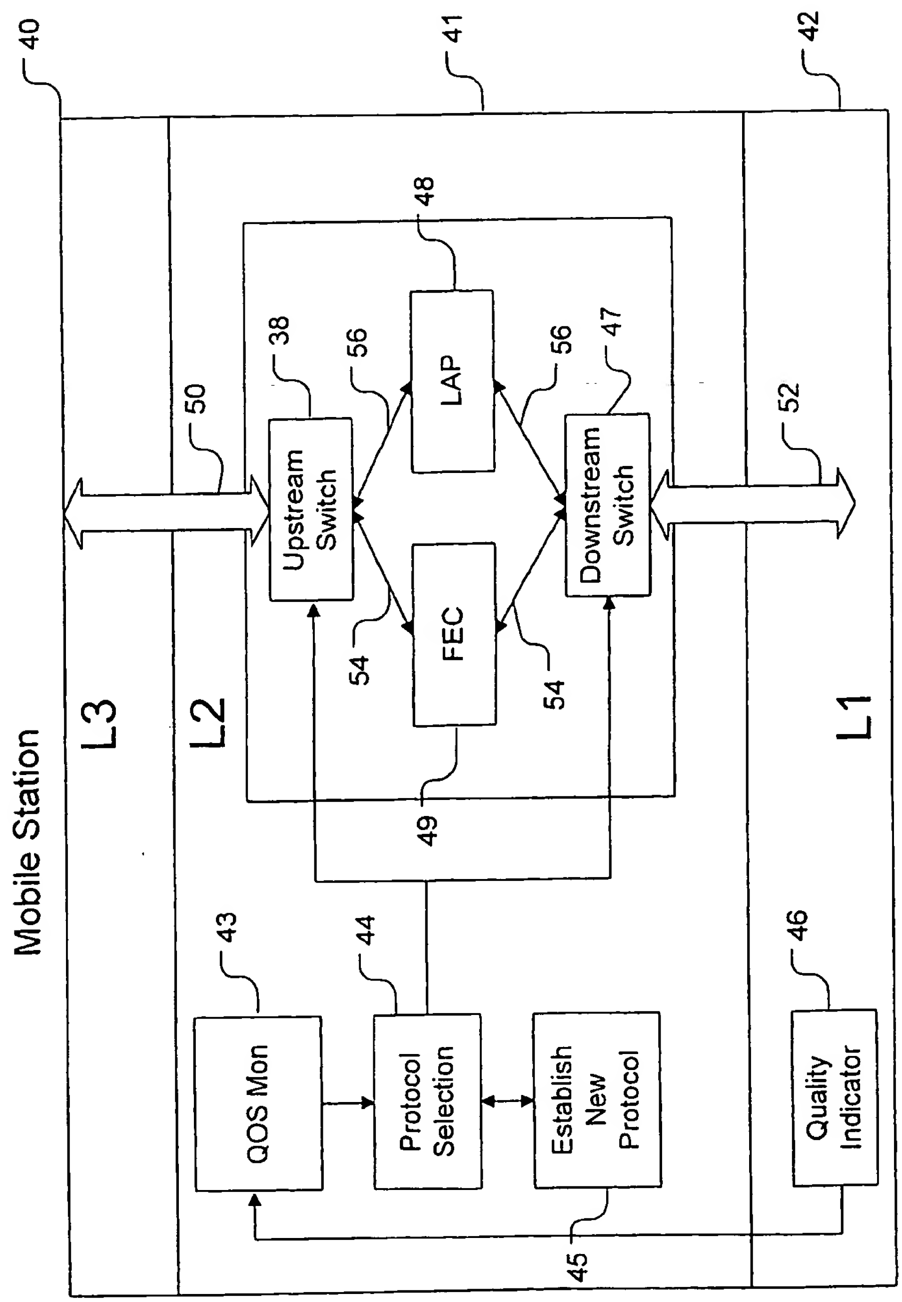

Mobile Station
L3
L2
L1
40
41
42
50
52
38
Upstream Switch
LAP
FEC
48
49
56
54
47
Downstream Switch
43
QOS Mon
44
Protocol Selection
Establish New Protocol
45
46
Quality Indicator

Fig. 3A

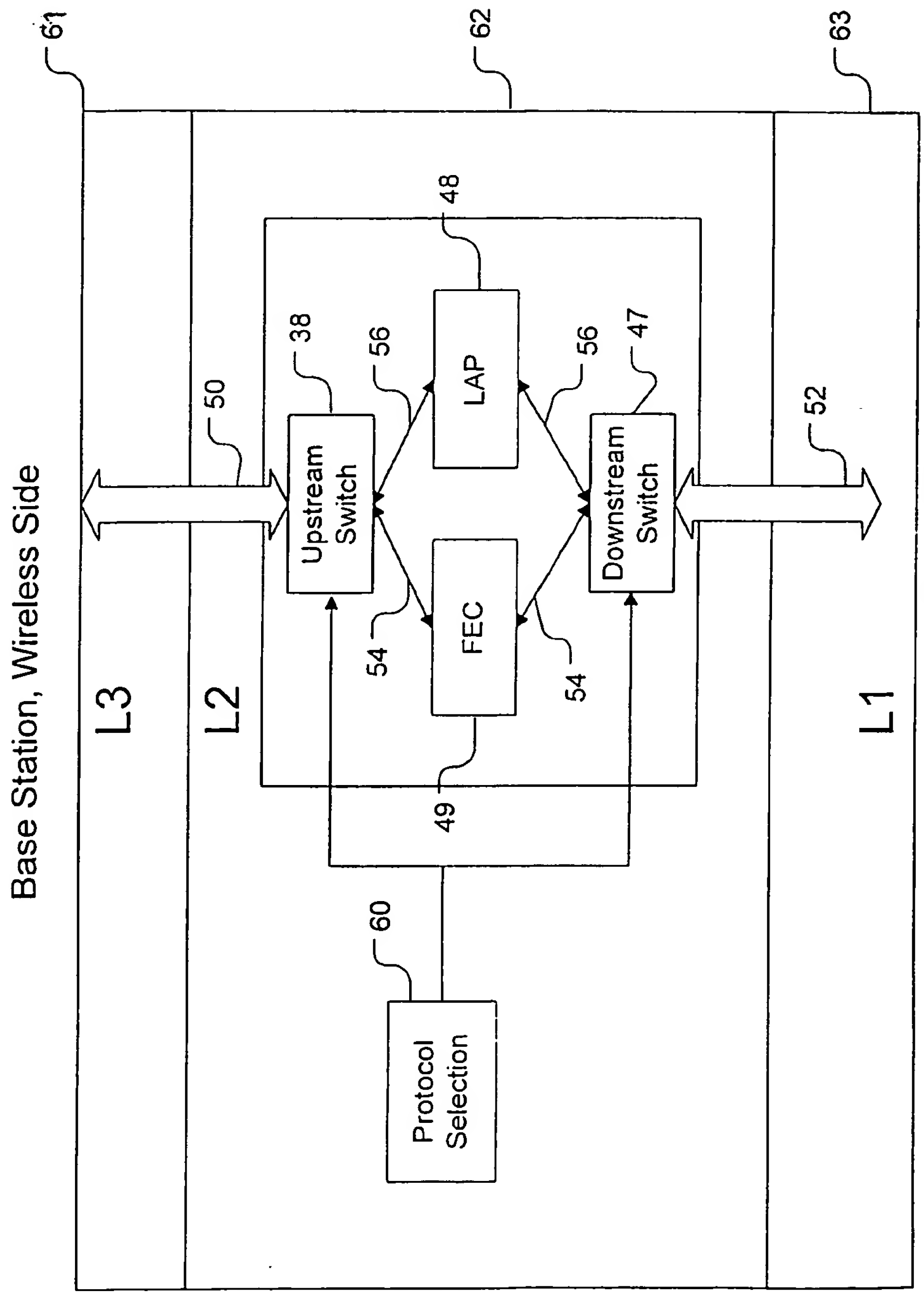

Base Station, Wireless Side
61
62
63
L3
L2
L1
50
52
38
Upstream Switch
47
Downstream Switch
48
LAP
49
FEC
56
56
54
54
60
Protocol Selection

Fig. 3B

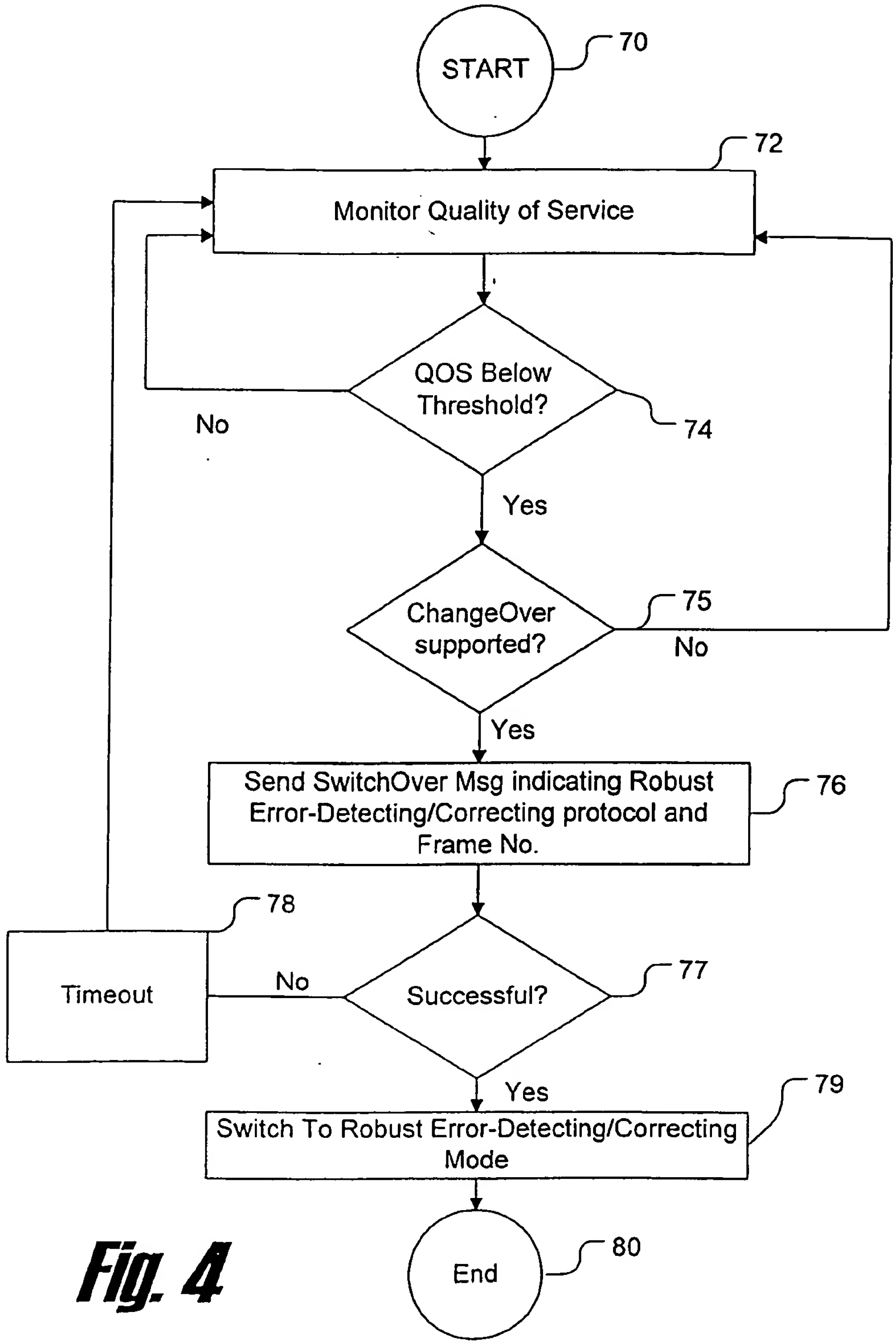
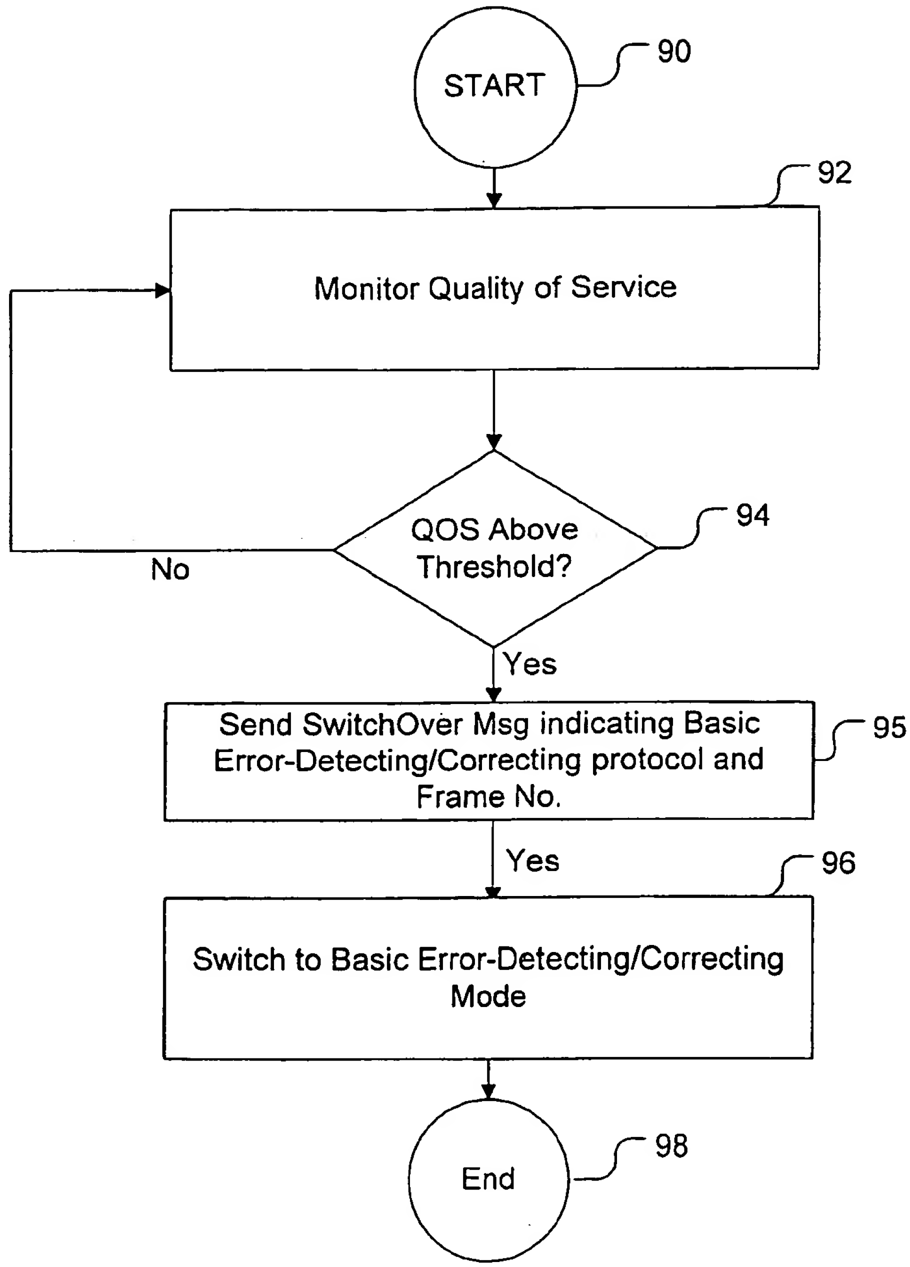
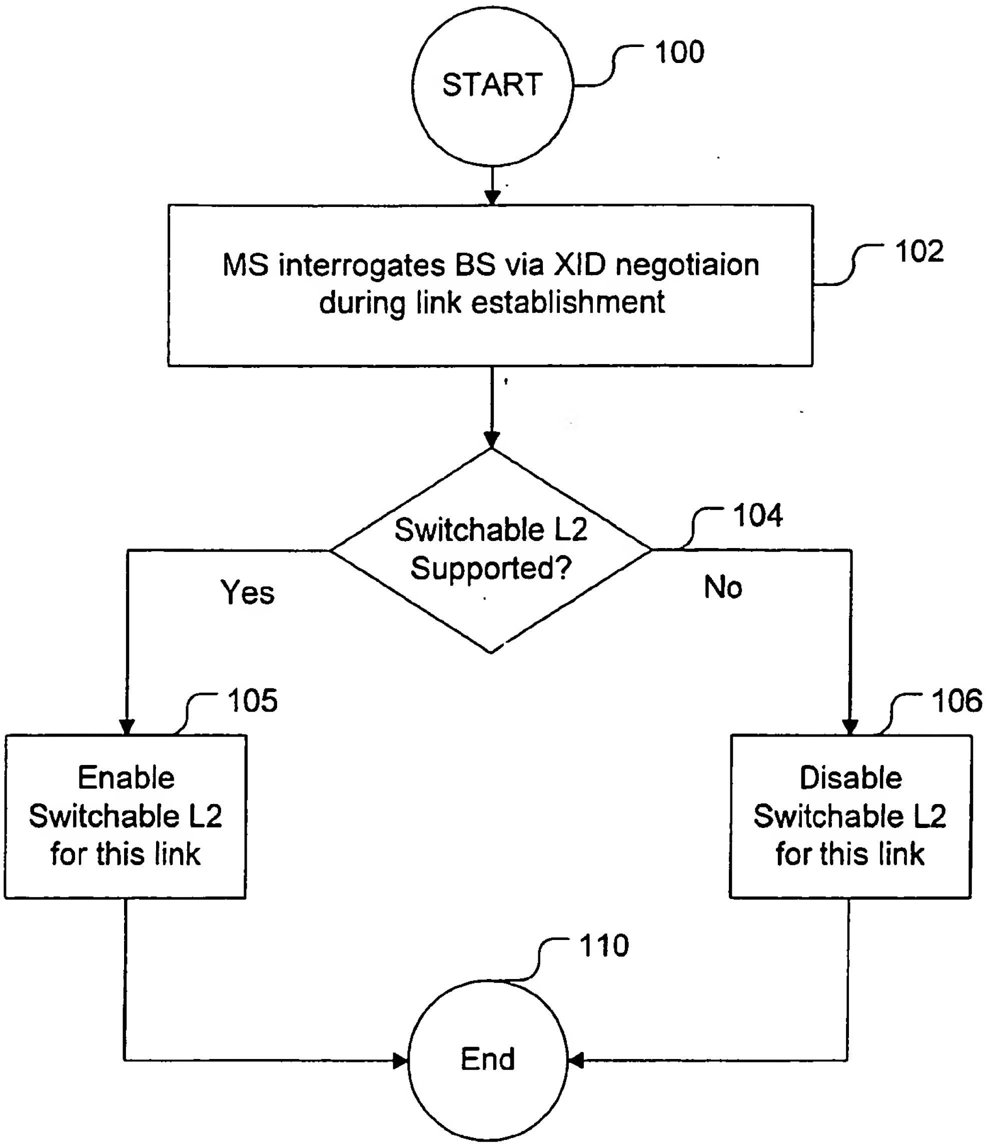

START
70
Monitor Quality of Service
72
QOS Below Threshold?
74
No
Yes
ChangeOver supported?
75
No
Yes
Send SwitchOver Msg indicating Robust Error-Detecting/Correcting protocol and Frame No.
76
Timeout
78
No
Successful?
77
Yes
Switch To Robust Error-Detecting/Correcting Mode
79
End
80

Fig. 4



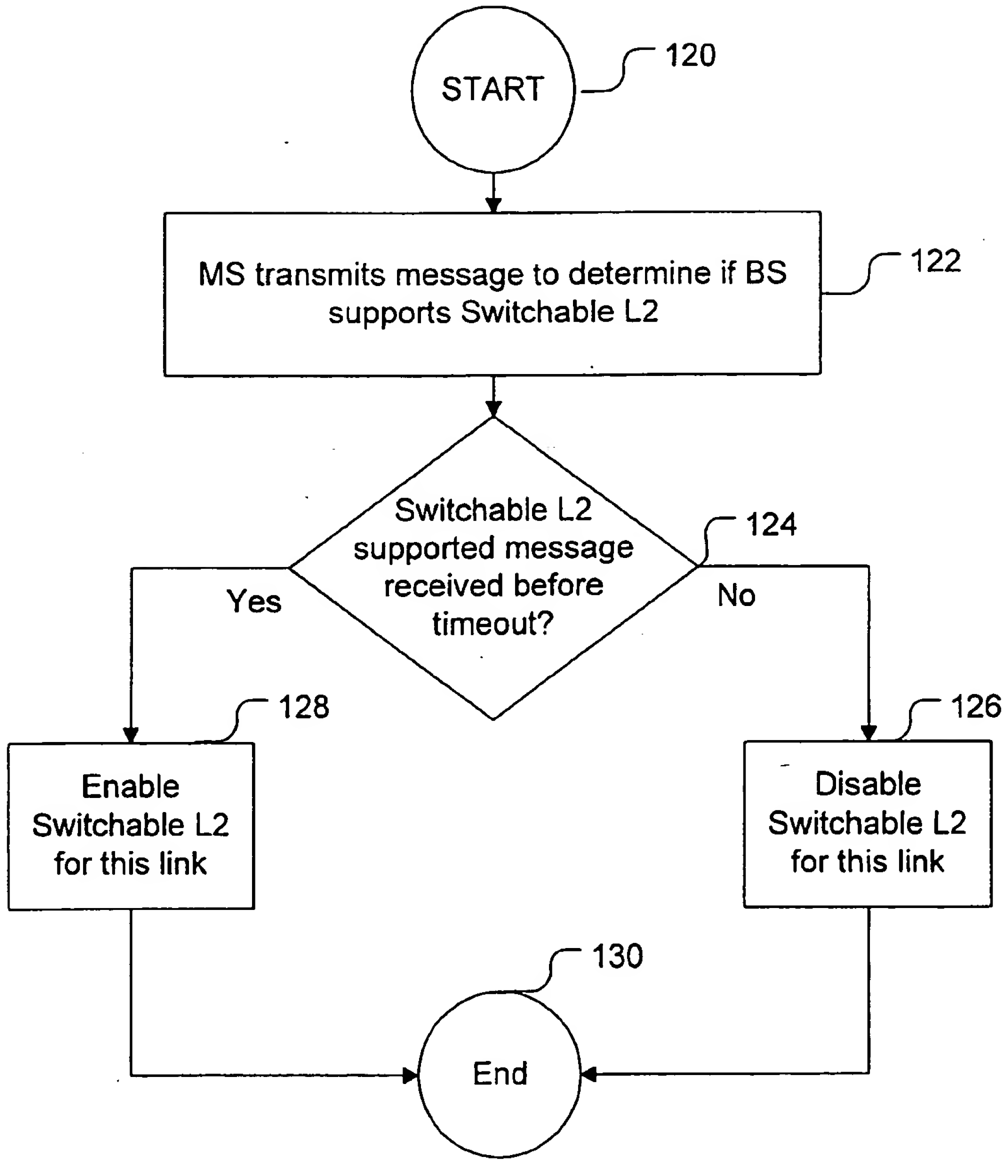
START
90
92
Monitor Quality of Service
QOS Above Threshold?
94
No
Yes
Send SwitchOver Msg indicating Basic Error-Detecting/Correcting protocol and Frame No.
95
Yes
96
Switch to Basic Error-Detecting/Correcting Mode
End
98

Fig. 5



Determine Support for Switchable L2 During XID Negotiation

Fig. 6



Determine Support for Switchable L2 via Message

Fig. 7

800

```
/* Table Preparation and Table Handling instructions
*/

Store vector of Signal Quality which would at a particular
dues in SQ_table (this would be the y-data on y vs x plot)

    IF (vector of CNR values corresponding to Signal
Quality values is to be      uniformly spaced)

    {

        store min_CNR_in_table, CNR_table_increment, and
max_CNR_in_table

    }

    ELSE (vector of CNR values corresponding to Signal
Quality values is not to       be uniformly spaced

    {

        store vector of CNR values corresponding to Signal
Quality values as        CNR_table

        store length_of_CNR_table

        (during    computations    one    will    access
min_CNR_in_table and                     max_CNR_in_table
via the appropriate addressing indices, which are,
  1 or, length_of_CNR_table, respectively, of CNR_table)

    } END IF
```

Fig. 8A

```
/* Initialization */

Select Allowable Interference Margin in dB,
Intf_Margin

/* Set the Interference Level to Zero */

Intf_level_est ← 0;
```

Fig. 8B

```
/* General Operation */

FOR each frame:

{

    Get instantaneous RSSI measurement for Frame (max
of early/middle/late),    RSSI

    Get instantaneous Signal Quality measurement for
frame, SQ

    Look up CNR_basedon_SQ corresponding to this SQ
from table. i.e.find value in SQ_table closest to SQ,
and determine the CNR associated with this SQ table
value. Call this value CNR_basedon_SQ. If SQ does not lie
within bounds of SQ_table, use association based on
closest value which is in the table.

    /* account for cases where RSSI does not lie
within tabulated bounds */

    IF (RSSI >= max_CNR_in_table)

      RSSI ← max_CNR_in_table

    ENDIF


    IF (RSSI <= min_CNR_in_table)

      RSSI ← min_CNR_in_table

    ENDIF
```

Fig. 8C

Continued from FIG. 8C

/* Filter difference between measured RSSI and RSSI perceived by SQ measurement

Intf_level_est ← (1-α) **Intf_level_est** + α (**RSSI** - **CNR_basedonSQ**)

/* α is a TBD parameter which is (1/2) to some integer power. 1/16 or 1/32 tend to work well in simulations */

/* X_{TBD} is a value TBD which relates to the variability in RSSI measurements and choice of small α---which limits the amount of averaging done. A small α, however, is necessary for quick transient responses. X_{TBD} should be small. */

IF (**Intf_level_est** > **Intf_margin** + X_{TBD})

Send 'Interference Detected from SQ' message to RME, along with value of **Intf_level_est**

ENDIF

END For each Frame 'FOR LOOP'

Fig. 8D

PERFORMANCE IMPROVEMENT OF INTERNET PROTOCOLS OVER WIRELESS CONNECTIONS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to a system and method for providing improved performance of Internet protocols over wireless connections, and more specifically, to a system and method for improving performance of Internet protocols by preventing TCP/IP congestion avoidance mode as a result of errors in the wireless connection.

2. Related Art

Current trends indicate a significant increase in the use of mobile computing devices such as auto PCs, personal digital assistants (PDAs) and the like. Accordingly, there is an increasing demand for fast and reliable wireless communication links to computer networks, such as the Internet.

The problem is that current communication techniques using TCP/IP protocol suites have proven to be troublesome when used in wireless networks. This is primarily due to the fact that TCP/IP was originally designed to be used with hard-wired or "fixed" transmission links rather than wireless radio links.

Specifically, the TCP/IP suite of protocols was designed for environments using highly reliable transmission media with very low bit error rates (BER). Thus, TCP/IP was designed with the assumption that the underlying physical connections used to transport the data were error-free. Consequently, TCP/IP assumes that low bandwidth conditions are caused by congestion, rather than problems related to the underlying transport media. Accordingly, TCP/IP responds to low bandwidth conditions (i.e. packet losses) by slowing down the transmission rate. This works well for wired networks but can be disastrous for wireless connections.

Typically, packet losses in wireless networks are not caused by congestion but are caused by high bit error rates due to weak and fading wireless transmission links. The quality of such wireless transmission links is affected by physical obstructions, weather conditions, atmospheric conditions, power failures, and the distance between cells (and density of cells) in a cellular network. In most cases, congestion avoidance (by stopping transmission for a period of time) during adverse conditions is an inappropriate response in a wireless network. Moreover, the resultant reduction in overall transmission rates during rapidly fading conditions only serves to further increase already existing bandwidth problems.

One of the built in error-correcting procedures used by TCP/IP is referred to as congestion avoidance mode. When packet losses are detected, TCP/IP enters into congestion avoidance mode, which causes an exponential reduction in the transmission rate. This procedure works well for fixed networks because the packet loss is almost always due to traffic congestion.

However, in a wireless network, where the packet loss is often due to rapidly fading conditions, a reduction in the transmission rate should be avoided at all costs. Entering into congestion avoidance mode only serves to increase problems and cause additional and unnecessary delays. Instead, during these conditions, throughput should be maintained and other forms of error correction techniques should be used.

Solutions to this problem have been proposed as is evidenced by the following publications: (1) R. Cáceres et al. "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 5, June 1995; (2) P. Karn, "The Qualcomm CDMA digital Cellular System," Proceedings of the USENIX Molile & Location-Independent Computing Symposium, pp. 35–39, August 1993; (3) Hbalakrishnan et. al "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," Wireless Networks, Vol. 1, No. 4, pp 469–481, 1995; (4) J Saltzer et al. "End-to-end Arguments in System Design," ACM Transactions on Computer System Design," ACM Transactions on Computer System (TOCS), Vol. 2, No. 4, pp. 277–288, 1984; and (5) B. R. Bandrinath et. al "Handling Mobile Clients: A Case for Indirect Interaction," Proceddings of the 4th Workshop on Workstation Operating Systems, pp. 91–97, October 1993.

However, these conventional solutions require changes to the link layer to provide error free service to the transport layer. Further, current link layer error detection and correction schemes introduce additional delays due to the retransmission of data. Thus, using techniques, such as the sliding window protocols as suggested by (1) above, can still trigger congestion avoidance mode in TCP/IP.

Another approach is to offer limited link layer recovery, leaving complete recovery to the transport layer, as suggested by (2) above. The problem with this solution is that it doesn't address TCP/IP congestion avoidance issue at all, it just delays its onset.

Yet another conventional approach, as suggested by (3) above, is to add a snoop agent to base station routing software. Using this approach, the snoop agent maintains a cache of unacknowledged base station-to-mobile station TCP packets. When a packet loss is detected (e.g. via duplicate acknowledgements or a local timeout), the snoop agent retransmits the lost packet, preventing congestion recovery by TCP. An acknowledgement from the mobile station allows the snoop agent to clean up its cache, update round trip time estimates, etc.

The problem is that these conventional approaches all require significant overhead in terms of software and design changes to current systems. For example, the snoop agent approach requires that IP multicast be implemented and used during cellular handoffs. Further, this solution requires that TCP be modified on the mobile station side. Still further, this and other conventional approaches violate protocol-layering principles by using information and messages in the transport layer for link layer purposes.

Accordingly, what is needed is a system and method for providing improved transmission rates using TCP/IP protocols over wireless networks that can be implemented without changing TCP/IP protocols and circumvents inappropriate occurrences of TCP congestion avoidance mode.

SUMMARY OF THE INVENTION

Accordingly, the present invention is directed toward a system and method for providing improved performance of Internet protocols over wireless networks. An advantage of the present invention is that it can be implemented entirely within the link layer of a protocol stack and does not affect any of the other layers. Another advantage of the present invention is that it responds appropriately to low-bandwidth conditions caused by weak and fading wireless connections by maintaining throughput and thereby circumvents inappropriate instances of TCP/IP congestion avoidance mode.

The present invention comprises at least one basic error-correcting/detecting service protocol and at least one robust error-detecting/correcting service protocol within the link layers of the mobile and base station's protocol stacks. Communications are initiated using a well known basic error-detecting/correcting protocol, such as Link Access Protocol (LAP). During data communications, a quality of service monitor constantly monitors the signal quality, via a signal quality indicator, generally coupled to the physical layer. An example of a signal quality indicator is a data logger that maintains a log of bit error rates during data communications. Another example of a quality indicator is a channel quality estimator that can be used to predict a degradation of signal quality.

When the quality of service monitor detects that the signal quality falls (or is about to fall), below a predetermined threshold, the mobile station attempts to switch to the robust error-detecting/correcting service protocol to be used during weak conditions. An example of an robust error-detecting/correcting protocol is forward error correction (FEC) protocol.

To accomplish the changeover, the mobile station first determines whether the base station supports the switchable protocol feature of the present invention. If the feature is not supported, then the changeover is temporarily disabled. In this fashion, the present invention is transparent and fully compatible with conventional systems. Generally, the determination of whether the base station supports the changeover feature of the present invention is accomplished during the link establishment phase.

If the changeover feature of the present invention is supported, a switchover message is sent from the mobile station to the base station directing it to switch to the robust error-detecting/correcting protocol on the next frame (or a specified frame number). The mobile station and the base station switch to the robust error-detecting/correcting protocol, which is used during weak signal conditions. If the quality of service rises above a second predetermined threshold, the basic error-detecting/correcting protocol is restored in both the mobile and the base stations.

BRIEF DESCRIPTION OF THE FIGURES

The present invention is described with reference to the accompanying drawings, wherein:

FIG. 1 is a diagram depicting a typical operation environment, according to an embodiment of the present invention.

FIG. 2 is a block diagram depicting portions of protocol stacks within a mobile station and a base station, according to an embodiment of the present invention.

FIG. 3A is a block diagram depicting components of the present invention within a mobile station, according to an embodiment of the present invention.

FIG. 3B is a block diagram depicting components of the present invention within a base station, according to an embodiment of the present invention.

FIG. 4 is a flowchart depicting a process that can be used to implement an embodiment of the present invention.

FIG. 5 is a flowchart depicting a process that can be used to switch back from an robust error-detecting/correcting protocol to an basic error-detecting/correcting protocol, according to the present invention.

FIG. 6 is a flowchart depicting a process that can be used to determine if a base station supports the changeover protocol feature of the present invention during XID negotiation.

FIG. 7 is a flowchart depicting a process that can be used to determine if a base station supports the changeover protocol feature of the present invention via a message sent during data communications.

FIGS. 8A–8D are example algorithms written is pseudocode that can be used for Signal Quality Statistic-based Interference Tracking in accordance with an embodiment of the present invention.

In the figures, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention is directed toward a system and method for providing improved performance using TCP/IP protocols over wireless networks that can be implemented entirely within the link layer of a protocol stack. The system and method of the present invention responds to poor quality channel conditions caused by weak and fading wireless connections by maintaining transmission throughput and preventing TCP/IP congestion avoidance mode.

FIG. 1 is a diagram depicting a typical operating environment according to a preferred embodiment of the present invention. A mobile station 7 is typically installed within a mobile unit such as an automobile or the like. In one embodiment, the mobile station 7 is a general-purpose computer system running an application program, such as a World Wide Web browser. In other embodiments the mobile station 7 is a PDA, an Auto PC, a pager, a cellular telephone, a navigational computing system, or the like.

As shown in FIG. 1, the mobile station 7 is performing end-to-end communications with the fixed station 2. The fixed station 2 is coupled to a computer network 1, such as the Internet. As shown, part of the communication path between the fixed station 2 and the mobile station 7 is maintained through a wireless communication link.

It should be noted that the Internet is just one example of a computer network that can be used with an embodiment of the present invention. In other embodiments, any type of computer network can be used. Accordingly, the use of the Internet in the examples presented herein, should not be construed to limit the scope and breadth of the present invention

In the examples presented below, cellular technology is used to implement the wireless communication links of the present invention. For example, in FIG. 1, the base stations 3A and 3B (generally, 3) and the mobile switching center 8 represent a portion of a typical cellular network. The base stations 3 send and receive radio signals to and from the mobile station 7. The mobile switching center 8 is coupled to the base stations 3 via standard telecommunication transmission lines. Likewise, the mobile switching center 8 is typically coupled to a public switched telephone network 9 via standard telecommunication transmission lines. Typically, a public switched network 9 is coupled to the Internet 1 at point-of-presence using high bandwidth telecommunication channels.

The cells 5 and 6 represent the range of the transceivers in each of the base stations 3A and 3B, respectively. As the mobile station 7 moves within cell 5, or between the cells 5 and 6, it transmits and receives data packets with the particular base station 3 associated with the current cell. Generally, the cells 5 and 6 overlap one another so that the mobile station 7 is always in radio communications with at least one base stations 3. Switching from one base station,

such as 3A to another base station, such as 3B, is referred to herein as handover.

Note that the cellular network shown in FIG. 1 is just one example of a technology that can be used to implement the wireless communications of the present invention. In other embodiments, different types of wireless technology can be used, such as low orbit or geosynchronous orbit satellite communications. In fact, any type of wireless technology can be used to provide the wireless communication links in accordance with the present invention.

FIG. 2 is a block diagram depicting details of the mobile station 7 and the base station 3 in accordance with one embodiment of the present invention. In particular, FIG. 2 depicts a portion of the protocol stacks 20 and 21 that are implemented via software and/or hardware within the mobile station 7 and the base station 3, respectively. The communications path indicator 24 depicts the logical communication flow through the various protocol layers within the protocol stacks 20 and 21.

As shown, the mobile station 7 comprises a protocol stack 20. The protocol stack 20 as depicted, is based on the Open System Interconnection (OSI) reference model. In this example, only those portions of the OSI reference model relevant to the present invention are shown. Accordingly, the protocol stack 20, in this example, comprises a physical layer (layer-1) 25, a data link layer or "link layer" (layer-2) 24, a network layer (layer-3) 23 and a transport layer 22. An application layer is depicted as the application module 30.

It should be noted that the layered models presented in the examples herein are simplified versions of actual layered protocol stacks. These simplifications are used for the sake of clarity and to distinctly point out the details of the present invention. For example, layers that are not relevant to the description of the present invention are not depicted. In addition, TCP/IP is depicted and described as a single layer, rather than four separate layers (i.e. L4 {Telnet or FTP or e-mail etc.}; L3 {TCP or UDP}; L2 {IP or ICMP or IGMP}; L1 {interface to whatever is underneath}.).

It is noted that TCP/IP is a protocol suite and is generally considered to be a 4-layer system consisting of a number of protocols at each of the four layers. It is, however, common practice to refer to the TCP/IP protocol suite as if it was a single layer.

This simplification is made to emphasize the point that changes to the TCP/IP are not required in accordance with a preferred embodiment of the present invention.

Referring now to FIG. 2, the shaded portions of the protocol stacks 20 and 21 are the only portions that are modified in accordance with a preferred embodiment of the present invention. In particular, the present invention is implemented by modifying only the link layer 24 of the mobile station's protocol stack 20 and the link layer 27 in the wireless side 18 of the base station's protocol stack 21.

Thus, unlike conventional solutions, which require extensive modifications, an advantage of the present invention is that TCP/IP is not modified to implement the performance improvement over wireless networks.

As described below, an embodiment of the present invention is implemented at the link layer. In general, the link layer takes data from the higher layers, creates data packets, and sends the packets out through the physical layer. In the opposite direction, the link layer receives packets from the physical layer, then combines the packets and sends data to the higher layers.

Thus, by only modifying the link layer 24, the present invention is transparent to higher layers (23, 22, and 27) and to the physical layer 25 and 26. In this fashion, application programs, such as the application program 30 and the operating system are not altered. Instead, as described below, the present invention is implemented entirely via firmware changes within network devices such as modems, cellular network communication devices and the like.

Each layer in the protocol stacks 20 and 21 offer a service to the layer above. These services are depicted in FIG. 2 as the double-sided arrows between the layers. For reasons of speed and efficiency, the physical layer 25 is typically implemented in hardware. .The link layer 24, network layer 23 and the TCP/IP protocol suite 22 are generally implemented in software. Typical services offered to the link layer 24 by the physical layer 25 are channel detection/selection; frame extraction and delivery to the link layer 24. The link layer 24 will perform an error detection and correction service, a re-assembly service and a delivery service to the network layer 23. The network layer 23 may offer an encryption/decryption service and a header compression/decompression service to the IP layer within the TCP/IP protocol suite 22. The TCP/IP protocol suite 22 offers a reliable data transport service to the application program 30. Examples of such protocol stacks are IS-707 CDMA Data Services; Cellular Digital Packet Data (CDPD); GSM Packet Radio Service (GPRS), and pACT.

The application module 30 represents an application program, such as a web browser or the like. Such application programs 30 generally run on top of the protocol stack. Examples of web browser application programs include Internet Explorer® and Netscape Navigator®.

The path indicator 24 depicts the physical communication path from the mobile station 7 to the base station 3 through the various protocol layers. As shown, the application program 30 communicates with the TCP/IP suite 22. Next, as indicated, the TCP/IP suite 22 communicates with the network layer 23, which is generally part of the protocol stack. The network layer 23, communicates with the link layer 24, which as stated, are generally implemented in software. The link layer 24 communicates with the physical layer 25, which as stated, is generally implemented in hardware.

Generally, data from the physical layer 25 is transmitted over a transmission link to a receiving device. The receiving device, having a similar protocol stack as the sending device, receives the data through the receiving device's physical layer. The path 24 in this example shows the conceptual data path from the physical layer 25 in the mobile station 7 to the physical layer 26 in the base station 3. The physical data path is depicted by the wireless radio communication link 19.

The base station 3 comprises a protocol stack 21. In this example, the base station 3 serves as a communication gateway between two end-to-end peers, namely the mobile station 7 and the fixed station 2. Accordingly, only two layers, the physical and link layers, are required for the gateway device 3.

In addition, the protocol stack 21 comprises a wireless side 18 and a wired side 17. The wireless side 18 communicates with the mobile station 7, via a wireless communications link 19. The wired side 17 communicates with the mobile switching center 8, via a hard-wired communications link 18.

As indicated, data enters the physical layer 26 and is passed to the link layer 27 on the wireless side 18. The shaded block of link layer 27 indicates that changes to this layer are required for a typical implementation of the present invention. Next, as indicated, the data is sent to the link layer

28 on the wired side of the protocol stack 21. Next, the data is passed to the physical layer 29, where it is transmitted over the wired telecommunications link 18 to the mobile switching center 8.

As stated, the highlighted blocks 24 and 27 indicate the location where modifications are made to existing systems in accordance with an embodiment of the present invention. In particular, modifications as described below, are preferably made to the link layers 24 and 27 in the mobile station 7 and the base station 3, respectively. No other changes to existing hardware and/or software are required.

FIG. 3A is a block diagram depicting components that can be used to implement an embodiment of the present invention in the mobile station 7. A portion of a protocol stack in the mobile station 7 is represented in FIG. 3A. In particular, the protocol stack comprises a network layer 40, a link layer 41, and a physical layer 42. In a preferred embodiment, the present invention is implemented by adding components to the link layer 41. Accordingly, as shown, no changes are required in any of the other layers in the protocol stack.

At least two service protocols are used to implement this example embodiment of the present invention. In particular, as shown, the present invention utilizes a link access protocol (LAP) 48 and a forward error correction (FEC) protocol 49. More generally however, the present invention utilizes at least one basic error-detecting/correcting protocol (such as LAP 48), and at least one robust error-detecting/correcting protocol (such as FEC 49) to achieve the performance improvements of TCP/IP as described herein.

In other embodiments, more than two service protocols are used. However, it is important to note that unlike current systems, which use a single service protocol, the present invention makes use of at least two different selectable service protocols. As described below, a particular service protocol is selected based on a predefined criteria associated with the wireless channel quality.

Before describing the details of this example embodiment of the present invention, it is important to note the general distinctions between basic error-detecting/correcting protocols and robust error-detecting/correcting protocols. In general, basic error-detecting/correcting protocols, such as LAP 48, add a small amount of redundant information to each data frame. This small amount of redundant information is used by the receiving system to detect the occurrence of errors. When errors are detected, the sending system is asked to re-transmit one or more data frame(s) associated with the error(s).

Robust error-detecting/correcting protocols, on the other hand, include a larger amount of redundant information with each data frame. This redundant information is sufficiently large to correct the data when errors that are detected. Accordingly, re-transmitting data frames are generally not required when robust error-detecting/correcting protocols, such as FEC 49, are used.

The benefit to using basic error-detecting/correcting protocols, such as LAP 48, rather than robust error-detecting/correcting protocols, such as FEC 49, is that the former requires less redundant information, thereby increasing the data payload carried on a channel.

Robust error-detecting/correcting protocols, such as FEC 49, are generally used for mission critical data that must be received on the first transmission. For example, robust error-detecting/correcting protocols are used to transmit telemetry data from spacecraft to mission control centers. In this environment, the re-sending of data is either impracticable or impossible. Robust error-detecting/correcting protocols are also used in noisy environments and when large bandwidth channels are available.

Current link layer implementations of communication protocol stacks using TCP/IP generally use a LAP error-detecting service protocol. LAP is a common full duplex, point-to-point bit synchronous data link control protocol. The application program 30 presents a data stream for transmission to TCP/IP 22. TCP/IP 22 will split this data into packets. These packets are modified by the addition of headers containing packet identification information. These packets are passed to the network layer 23, where the packet headers may be compressed, and the packet encrypted. The packet is passed the link layer 24, where it is divided into frames. How these frames are further modified depends on the type of link protocol used. If a basic error-detecting/correcting protocol is used (e.g. LAP) the frames are modified by the addition of headers containing frame identification information. If a robust error-detecting/correcting protocol is used (e.g. FEC), headers containing frame information are added and correction bits are inserted into the frame. Examples of such coding schemes are Hamming Code and Convolutional Code.

As stated, the link layer embodiment of the present invention comprises at least one selectable basic error-detecting/correcting protocol and at least one selectable robust error-detecting/correcting protocol. In this example, the robust error-detecting/correcting protocol is represented by the FEC module 49, and the basic error-detecting/correcting protocol is represented by the LAP module 48.

In operation, the present invention defaults to the LAP basic error-detecting/correcting protocol 48. Accordingly, the LAP module 48 is used during normal operating conditions when the channel quality is above a predetermined threshold. However, when the channel quality (or predicted channel quality) falls below a predetermined level, the present invention attempts to switch to the FEC module 49 to implement the forward error-correcting service protocol.

As described below, the switch to another service protocol can be achieved if the currently communicating base station 3 supports the "changeover feature," also referred to as the "switchable L2 feature," in accordance with the principles disclosed herein. Otherwise, the LAP protocol is maintained and operations continue using the LAP module 48. In this case, the performance improvements of the present invention will not be realized, but this feature makes the present invention compatible with conventional systems.

Referring back to FIG. 3A, the FEC module is coupled with a downstream switch 47 and an upstream switch 38. Similarly, the LAP module 48 is coupled with the downstream switch 47 and the upstream switch 38. The upstream and downstream switches 38 and 47 function to enable either the LAP module 48 or the FEC module 49 and control the data flow through the appropriate protocol module.

Thus, when the LAP module 48 is enabled, the data path indicated by the arrows 56 is active. Similarly, when the FEC module 49 is enabled, the data path indicated by the arrows 54 is active. The upstream and downstream switches 49 and 47 operate in concert to control the data flow. As shown, a protocol selection module 44 is coupled to the upstream and downstream switches 49 and 47. The protocol selector 44 controls when the switches 38 and 47 are activated to switch protocols.

The protocol selector 44 is coupled with a quality of service (QOS) module 43. The QOS module 43 is coupled with a quality of service indicator 46, which is generally coupled to the physical layer 42. As the name implies, the

quality of service indicator **46** is used to indicate a quality of service. The quality of service indicator can be of many forms as long as it indicates either a current or a predicted quality of service.

For example, the quality indicator **46** can be as simple as a data logger found in many cellular telephone transceivers. Generally, a data logger keeps track of bit error rates (BER) for data transmissions and is used as a debugging tool by service technicians.

In this example, the data logger **46**, or more generally, the quality indicator **46**, is used to track the bit error rate of data transmissions. This is monitored on a continual basis by the QOS module **43**. When the QOS module determines that the quality indicator is below a certain predefined threshold, it alerts the protocol selection module **44** to attempt to switch protocols.

As stated, many other means can be used as the quality indicator **46**. Using the above example of the bit error rates, the quality indicator reports a historical and current signal quality. In a preferred embodiment of the present invention, a predicted or future indication of signal quality is preferred. A future indication of signal quality is preferred to allow for additional time to correct errors during rapidly fading conditions.

For example, to predict a signal quality, signal channel quality estimates are derived from a measurement of the received signal power, retrieved from the physical layer **25**. This could for example, predict a fading signal quality before it occurs.

In one example embodiment, three separate channel quality estimation methods are used as follows:

1) Test T1: Signal-to-Noise-and-Interference-Ratio (SINR). Test T1 is used when a packet is not in error.
2) Test T2: Packet Error test (PER). Test T2 is used when a packet is in error.
3) Test T3: Average Signal strength (SNR). Test T3 is used for all packets.

In this fashion, by including all three methods T1, T2, T3, a large variety of channel failure mechanisms and impairment levels may be covered. For example, low levels of interference will degrade packet communications, whereas higher levels can completely obliterate them. By being responsive to gradual channel degradation (as one would largely be with estimation based on 'error-free' received packets T1), countermeasures can be taken before the overall communication quality becomes 'notably bad'. In addition, by immediately detecting, with low false alarm probability, sudden catastrophic interference events, countermeasures T2 can be taken to significantly diminish the impact of catastrophic events, so that they are barely perceived by the application **30**. Interference may not be the culprit for a significant number of packet errors: a diminishing average signal level may be. The average signal strength test T3 makes this determination.

When many packet errors occur, the SINR test T1 would be seldom functional, because it only operates on error-free packets. For the large PER case, one solution would be to cross-check the average PER statistic against the average signal level statistic. If the measured FER is much larger than a projected FER based on the average signal level (say, using a plot of PER vs. SNR for a Rayleigh fading channel), then interference is probably present.

When the PER is large, adding extra error correction capability won't be of assistance, so the "average-PER vs. average-signal-level test" is oriented more towards determining whether a channel switch, base station reassignment,

or exponential channel back-off should be performed. Therefore only details for the SINR test T1 are provided.

The SINR test T1 forms an SINR estimate by subtracting an estimate of the interference and noise power (in dB) from the measured signal power. The SINR is then be mapped to BER and/or PER using previously tabulated BER (or PER) vs. SNR characteristics (for the communication method and channel). This test is used determine whether the channel impairment is due to low signal levels (caused by Rayleigh fading or building shadowing, or propagation loss, for example) or interference (caused by channel congestion for example). This determination is required to allow the correct recovery action (e.g. protocol switch or congestion avoidance) to take place. The SINR statistic in itself is sufficient to determine what grade of service could be supported with a particular error correction strategy. If the SINR, (or projected PER or BER, or average SINR) drops below an allowable threshold, then a request can be made for increased error correction capability (i.e. FEC), a better channel, or a handover to a different base station.

Example: T1 Test SINR: SINR, PER, and BER Estimation for Packets Not in Error

Measurements

1. Signal+Noise+Interference Estimate: RSSI (Relative Signal Strength Indication) measurement. Available from the physical layer **25**, either from the RF circuitry or the demodulator circuitry
2. Noise+Interference Power Estimate: Average of the squared detection residuals (received symbol-detected symbol) for each received symbol in a packet.

This averaged statistic (mean-squared detection error) is generally compiled by the physical layer **25**, and is accessible by the higher layers.

Calculations

__SINR_in_dB=RSSI_in_dB-(estimated noise+interference power) in dB

Justification for the above statistic

(S+N+I)/(N+I)=S/(N+I)+I+RSSI→/(estimated noise+interference power)=1+SINR

SINR=RSSI/(estimated noise+interference power)−1

Typically, if the data is received without error, SINR>>1, which implies

SINR∓RSSI/(estimated noise+interference power), or, equivalently,

SINR_dB=RSSI_in_dB-(estimated noise+interference power)_in_dB

Significance of SINR in dB

When the SINR_in_dB begins to get small, the channel quality will become questionable for the current level of error protection, and Forward Error Correction should be requested.

Conversion to BER

A lookup table maps SINR_in_dB to expected BER, using a standard BER vs. SINR characteristic curve as its reference. Consulting this table allows a prediction of what the operating BER is, even when no bits are in error. BER vs. SINR curves are standard performance data calculated by Systems Engineers during the design process, and recorded by Field and Production Engineers for production parts. The mapping process is valid because that over a single packet, the signal level is relatively constant—even in a wireless environment—although it may not be constant from packet to packet.

Wireless communication standards such as PHS (Japan), DECT (Europe), and IS-95 (U.S.) share this characteristic.

This allows the use of BER vs. S[I]NR characteristics based on AWGN (additive white gaussian noise) rather than Rayleigh fading. This is advantageous because it leads to much higher levels of statistical significance with lesser numbers of packets. Rayleigh fading characteristics are averaged over an ensemble of fading power to form an average performance. This implies that many packets (and signal levels) must be spanned for them to have significance. (Without statistical significance, one cannot take any actions based on the data with a large degree of certainty that the data has been interpreted correctly).

Conversion to Packet error Rate PER

The SINR_dB can also be mapped directly into the packet error rate (PER) using PER vs. SINR statistics. This mapping can be done with a lookup table, similar to the BER conversion described above (if the channel SINR is constant over a frame interval of time).

Another conversion (which utilizes the BER estimate) is

PER=1-(1-BER)^ numsymbols_in_burst

Note that these calculations allow one to predict the frame/packet error rate accurately, despite the fact that statistics are taken only on frames not in error. This method has a much higher resolution than counting the number of packet errors, because it takes a large number of packets to get statistically sufficient numbers for the packet error rate.

Rationale for use only with packets not in error

The squared detection residuals used to form this estimate are not an accurate measure of the noise+interference power when there are bit errors in the frame. The bit errors bias the residuals. Residuals should only contain noise and interference contributions if they are to be used to estimate power levels of those sources.

Modifications to account for measurement errors etc.

Various uncertainties will arise in the measurement process. Sources of such uncertainties include manufacturing non-uniformity [primarily in RF parts], temperature of operation, tracking errors, or RSSI measurement accuracy. In a practical implementation, a mechanism must be invoked to desensitize the method to such uncertainties; otherwise, erroneous SINR and PER estimates, along with subsequent false alarms could result. For this reason, in practical applications, one preferably does not directly map the SINR_in_dB to a PER or BER figure. Instead, an adjustment for 'worst-case measurement errors' is made, and the resulting statistic is used during the FER or BER mapping. This statistic is calculated using the formula:

worst_case_SNR_in_dB=SINR_in_dB+error_margin_in_dB

where error margin_in_dB can be set to several dB, and yet detect degraded channels/interference quite well. However, the more the margin, the less sensitive the algorithm is in detecting channel impairments and interference.

An alternative approach to directly computing the SINR (which achieves similar aims) is to track the interference level. FIGS. 8A–8D are example algorithms (in pseudocode) that can be used for Signal Quality Statistic-based Interference Tracking. The use of this pseudocode to implement these algorithms would be apparent to persons skilled in the relevant art(s). In this example, Signal Quality is defined as the sum of detection residuals. Signal Quality vector is defined as the Signal Quality evaluated at a number of different SNRs, stored one entry in the vector per SNR value.

It is noted that other means, in addition to the example described above, can be used by the QOS monitor **43** to indicate a current or to predict a future signal quality. Accordingly, the examples used herein should not be construed to limit the scope and breadth of the present invention.

Referring back now to FIG. 3A, the protocol selection module **44** is coupled to an establish new protocol (ENP) module **45**. The ENP module **45** determines if the current base station **3** supports the switchable L2 feature of the present invention. If the changeover feature is supported, the ENP **45** operates to establish a new protocol with the base station **3**. Examples of procedures that can be used to determine if the base station **3** supports the changeover feature, are described below with reference to FIGS. **5** and **6**.

Once a new protocol has been established with the base station **3**, the ENP **45** sends a signal to the protocol selection module **44**. In response to this signal, the protocol selection module **44** switches protocols by sending the appropriate signals to the upstream and downstream switches **38** and **47**, simultaneously.

It should be noted that the organization and the description of the separate modules shown in FIG. 3A are for exemplary purposes only to distinctly point out and describe the features and functions of the present invention. Many other organizations or modules are possible, as would be appreciated by persons skilled in the relevant art(s). Accordingly, the use of the modules for describing the general functionality of the present invention should not be construed to be limiting.

FIG. 3B is a block diagram depicting components that can be used to implement an embodiment of the present invention in base station 3. A portion of a protocol stack in the base station 3 (wireless side **18**) is represented in FIG. 3B. In particular, the protocol stack comprises the network layer **61**, the link layer **62**, and the physical layer **63**. In a preferred embodiment, the present invention is implemented by adding components to the link layer **62** on the wireless side **18** of the base station **3**. Accordingly, as shown, no changes are required in any of the other layers in the protocol stack.

As shown, the components that are added to link layer **62** of the base station **3** are the same components as described above with respect to the mobile station **7**. However, in this example, not all of the components used in the mobile station **7** are used in the base station **3**. In this example, the base station **7** lacks a QOS monitor, a quality indicator and an establish new protocol module. These modules are not included because the base station **3**, in this example, does not monitor signal quality. Instead, in this example, the base station **3** only responds to a switchover message from the mobile station **7**. As described below, when the base station **3** receives a switchover message it switches to the appropriate protocol (**49** or **48**).

However, in other embodiments, the base station **3** can monitor signal quality in much the same manner as described above with respect to the mobile station **7**. In that case, the components that are missing from FIG. 3B are added, such that FIG. 3B looks similar or exactly the same as FIG. 3A. The choice of functionality implemented within the base and mobile stations, **3** and **7**, depend on each specific implementation of the present invention.

In this example, however, the base station **3** receives a switchover message from the mobile station **7** to switch protocols. When this occurs, the protocol selection module **60** enables the protocol specified by the switchover message. All of the remaining components shown in FIG. 3B function the same as the associated components described above in FIG. 3A.

FIG. 4 is a flowchart depicting a process that can be used to implement the present invention. In particular, it is assumed that the default basic error-detecting/correcting protocol (such as LAP 48) is active. Typically, communications are established between a base station 3 and a mobile station 7 using the default basic error-detecting/correcting protocol.

The process begins with step 72. In step 72, the process monitors the quality of service. As stated, this can be performed by monitoring a quality indicator 46, such as a bit error rate logger coupled to the physical layer. In another example, a quality of service indicator can be achieved by predicting future signal quality using the methods described above. Next, in step 74, the process determines whether the quality of services is below a predetermined threshold. If not, the process loops back to step 72, where the quality of service is monitored on a continual basis, as indicated by the loop 72-74. If the quality of service falls below the predetermined threshold, control passes to step 75.

In step 75, the process determines whether the changeover feature of the present invention is supported. If it is not, control passes back to step 72, as indicated. Methods that can be used to determine if the changeover feature of the present invention is supported are described below.

If it is determined that the changeover feature is supported, control passes to step 76. In step 76, the process sends a switchover message to the base station 3 to indicate that a protocol switch to the robust error-detecting/correcting protocol should occur at the next, (or a specified) frame number. Next, in step 77 the process determines if the protocol switch was successful.

Several methods can be used to determine if the protocol switch in the base station 3 is successful. For example, using an unacknowledged scheme, it is always assumed that the switchover message was received and that the switchover to the new protocol is successful. Thus, using this scheme, control passes directly to step 79 (in fact, step 77 can be omitted altogether). It is noted that if this assumption is incorrect, the mobile station 7 will be using a robust error-detecting/correcting protocol and the base station 3 will be using a basic error-detecting/correcting protocol. This will cause a protocol failure, which will force a link-reestablishment to occur between the mobile station and the base station. The link-reestablishment procedure will serve to return both the mobile station 7 and the base station 3 to the default basic error-detecting/correcting protocol.

If an acknowledged scheme is used, step 77 determines if an acknowledged message is received within a predetermined time interval. If an acknowledged message is received, control passes to step 79. If the message is not received within a predetermined time interval, control passes to step 78, where after a predetermined time-out period, control passes back to step 72, where the process monitors the quality of service again. This process can continue until either an acknowledged message is received, or the number of attempts have exceeded a predetermined threshold and the process returns with an error indication (not shown).

As indicated by step 79, upon a successful response from step 77, the process switches modes from the basic error-detecting/correcting protocol to the robust error-detecting/correcting protocol, as described above. Next, as indicated by step 80, the process ends.

FIG. 5 is a flowchart depicting a process that can be used to switch back from a robust error-detecting/correcting protocol to a basic error-detecting/correcting protocol. The process begins with step 92, where the quality of service indicator is monitored. Next, in step 94, the process determines if the quality of service has risen above a predetermined threshold value.

Note that in a preferred embodiment, the value used for the predetermined threshold from step 94 is not the same as the value used for the predetermined threshold from step 74. Instead, a somewhat lower error rate should be used to switch back to the basic error-detecting/correcting protocol to avoid excessive mode switches. For example, suppose a bit error rate of 10% is selected for the threshold value in step 74. In this case, a lower error-rate (i.e. 5%), should be used for the threshold value to switch back to the basic error-detecting/correcting protocol.

Next, as shown in step 95, the process sends a switchover message to the base station 3 to indicate that a protocol switch to the basic error-detecting/correcting protocol should occur at the next, (or a specified) frame number.

Next, as indicated by step 96, the switch back to the basic error-detecting/correcting protocol is performed. As shown, in this example, there is no need to determine whether the receiving base station supports the changeover feature of the present invention. This is so, because in this example, the system had already switched from the basic error-detecting/correcting to the robust error-detecting/correcting protocol. Therefore, because the process depicted in FIG. 5 had previously been executed, it is known that the current receiving system supports the changeover feature.

FIG. 6 is a flowchart depicting a process that can be used to determine if a base station 3 supports the changeover feature of the present invention. The process begins with step 102. In step 102, the mobile station 7 interrogates the base station 3 to determine whether such support is implemented. Preferably, this is accomplished with the use of a new predefined message designed for this purpose. The message is preferably sent during the XID negotiation phase during link establishment.

Next, as indicated by step 104, control passes to step 105, if the switchable L2 protocol feature is supported. In step 105, the process enables the feature for the current link. This can be accomplished for example, by simply setting a bit in a register, or more generally, setting a BOOLEAN flag. This flag is checked, for example during step 75 in FIG. 4, to determine if the current base station supports the changeover feature.

The opposite procedure is performed in step 106, if it is determined that the changeover feature is not supported. That is, the register bit is cleared in this case. Next, as indicated by step 110, the process ends.

FIG. 7 is a flowchart depicting a process that can be used to determine if a base station 3 supports the switchable L2 protocol feature of the present invention. In this example, instead of making this determination during XID negotiation, it is made just before switching protocols (i.e. when a quality of service problem is detected).

The process begins with step 122. In step 122, the mobile station 7 sends a predetermined message to the base station 3 to determine if the changeover feature is supported. One method that can be used to accomplish this task is to use the Sync or Paging channel available on CDMA devices, for example.

Next, as indicated by step 124, the process waits for a predetermined time interval to receive a message back from the base station 3. If a message is not received within the predetermined time interval, or a message is received indicating no support for the feature, control passes to step 126. In step 126 the changeover feature is disabled. If a message is received that indicates support of the changeover feature,

control passes to step **128**, where the feature is enabled. Next, as indicated by step **130**, the process ends.

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A method for use by a mobile station to improve performance over wireless connections between a base station and the mobile station, said mobile station having at least two service protocols comprising a basic error-detecting/correcting protocol and a robust error-detecting/correction protocol different from the basic error-detecting/correcting protocol, said method comprising the steps of:

establishing a communication link with the base station using the basic error-detecting/correcting protocol over a transmission channel;

monitoring a signal quality of the transmission channel;

determining if the signal quality falls below a first predetermined threshold;

sending a switchover message to the base station to switch from the basic error-detecting/correcting protocol to the robust error-detecting/correcting protocol, if the signal quality falls below the first predetermined threshold; and

switching to the robust error-detecting/correcting protocol.

2. The method of claim 1, further comprising the steps of:

sending a switchover message to the base station to switch from the robust error-detecting/correcting protocol to the basic error-detecting/correcting protocol, if the signal quality rises above a second predetermined threshold; and

switching to the basic error-detecting/correcting protocol.

3. The method of claim 1, wherein the basic error-detecting/correcting protocol is a link access protocol.

4. The method of claim 1, wherein the robust error-detecting/correcting protocol is a forward error correcting protocol.

5. The method of claim 1, further comprising the steps of:

querying the base station to determine if the base station supports the at least two service protocols; and

performing said sending and switching steps only if the base station supports the the at least two service protocols.

6. The method of claim 5, wherein said querying step is performed using XID negotiation during said establishing step.

7. The method of claim 5, wherein said querying step is performed by sending a message to the base station and then waiting for an acknowledgement from the base station to indicate support.

8. The method of claim 1, wherein said step of monitoring the signal quality is performed by monitoring a bit error rate of the communication link.

9. The method of claim 1, wherein said step of monitoring the signal quality is performed by predicting a future change in signal quality.

10. The method of claim 9, wherein said step of predicting a future change in signal quality is accomplished by applying an estimate of signal to noise and interference ratio (SINR).

11. The method of claim 10, wherein said step of applying an estimate of SINR includes subtracting said estimate of the interference and noise power from a measured signal power.

12. The method of claim 11, wherein said SINR is converted into a bit error rate.

13. The method of claim 1, further comprising the step of receiving a switchover acknowledgement from the base station in response to said sending step.

14. The method of claim 1, further comprising the step of querying the base station to determine if the base station supports the at least two service protocols, wherein said querying step occurs between said determining step and said sending step.

15. The method of claim 1, wherein said switchover message includes a parameter indicating the service protocol.

16. The method of claim 1, wherein said switchover message includes a parameter indicating a frame number.

17. The method of claim 16, wherein said switching step occur after a frame having the frame number.

18. A system for providing improved performance over wireless connection comprising:

a mobile station having a protocol stack with a link layer comprising:

at least two service protocols comprising at least one basic error-detecting/correcting protocol and at least one robust error-detecting/correction protocol different from the at least one basic error-detecting/correcting protocol;

a quality of service monitor for determining a signal quality of a transmission channel: and

a protocol selector coupled to said at least one basic error-detecting/correcting protocol and said at least one robust error-detecting/correcting protocol and said quality of service monitor, for selecting among said at least one basic error-detecting/correcting protocol and said at least one robust error-detecting/correcting protocol in accordance with said quality of service monitor; and

a base station having a protocol stack with a link layer comprising:

at least two service protocols comprising at least one basic error-detecting/correcting protocol and at least one robust error-detecting/correction protocol different from the at least one basic error-detecting/correcting protocol; and

a protocol selector coupled to said at least one basic error-detecting/correcting protocol and said at least one robust error-detecting/correcting protocol and said quality of service monitor, for selecting among said at least one basic error-detecting/correcting protocol and said at least one robust error-detecting/correcting protocol in accordance with a switchover message from said mobile station.

19. The system of claim 18, wherein said protocol selector in said mobile station switches from the at least one basic error-detecting/correcting protocol to the at least one robust error-detecting/correcting protocol when said quality monitor indicates a degradation in said signal quality.

20. The system of claim 18, wherein said protocol selector in said mobile station switches from the at least one robust error-detecting/correcting protocol to the at least one basic error-detecting/correcting protocol when said quality monitor indicates an improvement in said signal quality.

21. The system of claim 18, wherein said quality of service monitor is coupled with a quality of service indicator coupled to the physical layer in the protocol stack.

22. The system of claim 21, wherein said quality of service indicator is a data logger that tracks a bit error rate of data communications.

23. The system of claim 21, wherein said quality of service indicator is a measure of a received signal power that is used for predicting a future signal quality.

24. The system of claim 23, wherein said received signal power is derived from relative signal strength indication measurement.

25. A mobile station configured to improve performance over wireless connections between a base station and the mobile station, the mobile station comprising:

at least two service protocols comprising a first error-detecting/correcting protocol and a second error-detecting/correction protocol different from the first error-detecting/correcting protocol;

a link establishment module for establishing a communication link with the base station using the first error-detecting/correcting protocol over a transmission channel;

a quality of service monitor for determining a signal quality of the transmission channel;

a transmitter for sending a switchover message to the base station to switch from the first error-detecting/correcting protocol to the second error-detecting/correcting protocol if the signal quality passes the first predetermined threshold; and

a protocol selector for switching to the second error-detecting/correcting.

26. The mobile station of claim 25, wherein said transmitter sends a switchover message to the base station to switch from the second error-detecting/correcting protocol to the first error-detecting/correcting protocol if the signal quality passes a second predetermined threshold, and wherein the protocol selector switches to the first error-detecting/correcting protocol.

27. The mobile station of claim 25, wherein the first error-detecting/correcting protocol is a link access protocol.

28. The mobile station of claim 25, wherein the second error-detecting/correcting protocol is a forward error correcting protocol.

29. The mobile station of claim 25 further comprising a query module for querying the base station to determine if the base station supports the at least two service protocols, wherein the switchover message is transmitted only if the base station supports the at least two service protocols.

30. The mobile station of claim 29, wherein the query module is a part of the link establishment module and uses XID negotiation.

31. The mobile station of claim 29, wherein the query module sends a message to the base station and then waiting for an acknowledgement from the base station to indicate support.

32. The mobile station of claim 25, wherein the quality of service monitor monitors a bit error rate of the communication link.

33. The mobile station of claim 25, wherein the quality of service monitor predicts a future change in signal quality.

34. The mobile station of claim 33, wherein the quality of service monitor applies an estimate of signal to noise and interference ratio (SINR) to predict the future change in signal quality.

35. The mobile station of claim 34, wherein the quality of service monitor subtracts said estimate of the interference and noise power from a measured signal power.

36. The mobile station of claim 35, wherein said SINR is converted into a bit error rate.

37. The mobile station of claim 25 further comprising a receiver for receiving a switchover acknowledgement from the base station in response to the transmitter sending the switchover message.

38. The mobile station of claim 25 further comprising a query module for querying the base station to determine if the base station supports the at least two service protocols, wherein the query module queries after determining that the signal quality has fallen below the first predetermined threshold, but before sending the switchover message.

39. The mobile station of claim 25, wherein said switchover message includes a parameter indicating the service protocol.

40. The mobile station of claim 25, wherein said switchover message includes a parameter indicating a frame number.

41. The mobile station of claim 40, wherein said switching step occur after a frame having the frame number.

42. The mobile station of claim 25, wherein the first error-detecting/correcting protocol is less robust than the second error-detecting/correcting protocol, and wherein the transmitter sends the switchover message if the signal quality fails below the first predetermined threshold.

43. The mobile station of claim 25, wherein the first error-detecting/correcting protocol is more robust than the second error-detecting/correcting protocol, and wherein the transmitter sends the switchover message if the signal quality rises above the first predetermined threshold.

44. A method for use by a mobile station to improve performance over wireless connections between a base station and the mobile station, said mobile station having at least two service protocols comprising a first error-detecting/correcting protocol and a second error-detecting/correction protocol different from the first error-detecting/correcting protocol, said method comprising the steps of:

establishing a communication link with the base station using the first error-detecting/correcting protocol over a transmission channel;

monitoring a signal quality of the transmission channel;

determining if the signal quality falls below a first predetermined threshold;

sending a switchover message to the base station to switch from the first error-detecting/correcting protocol to the second error-detecting/correcting protocol if the signal quality passes the first predetermined threshold; and

switching to the second error-detecting/correcting protocol.

45. The method of claim 44, further comprising the steps of:

sending a switchover message to the base station to switch from the second error-detecting/correcting protocol to the first error-detecting/correcting protocol if the signal quality passes a second predetermined threshold; and

switching to the first error-detecting/correcting protocol.

46. The method of claim 44, wherein the first error-detecting/correcting protocol is a link access protocol.

47. The method of claim 44, wherein the second error-detecting/correcting protocol is a forward error correcting protocol.

48. The method of claim 44, further comprising the steps of:

querying the base station to determine if the base station supports the at least two service protocols; and

performing said sending and switching steps only if the base station supports the the at least two service protocols.

49. The method of claim 48, wherein said querying step is performed using XID negotiation during said establishing step.

50. The method of claim 48, wherein said querying step is performed by sending a message to the base station and then waiting for an acknowledgement from the base station to indicate support.

51. The method of claim 44, wherein said step of monitoring the signal quality is performed by monitoring a bit error rate of the communication link.

52. The method of claim 44, wherein said step of monitoring the signal quality is performed by predicting a future change in signal quality.

53. The method of claim 52, wherein said step of predicting a future change in signal quality is accomplished by applying an estimate of signal to noise and interference ratio (SINR).

54. The method of claim 53, wherein said step of applying an estimate of SINR includes subtracting said estimate of the interference and noise power from a measured signal power.

55. The method of claim 54, wherein said SINR is converted into a bit error rate.

56. The method of claim 44, further comprising the step of receiving a switchover acknowledgement from the base station in response to said sending step.

57. The method of claim 44, further comprising the step of querying the base station to determine if the base station supports the at least two service protocols, wherein said querying step occurs between said determining step and said sending step.

58. The method of claim 44, wherein said switchover message includes a parameter indicating the service protocol.

59. The method of claim 44, wherein said switchover message includes a parameter indicating a frame number.

60. The method of claim 59, wherein said switching step occur after a frame having the frame number.

61. The method of claim 44, wherein the first error-detecting/correcting protocol is less robust than the second error-detecting/correcting protocol, and wherein said sending step sends the switchover message if the signal quality falls below the first predetermined threshold.

62. The method of claim 44, wherein the first error-detecting/correcting protocol is more robust than the second error-detecting/correcting protocol, and wherein said sending step sends the switchover message if the signal quality rises above the first predetermined threshold.

63. A base station configured to improve performance over wireless connections between the base station and a mobile station, the base station comprising:

at least two service protocols comprising a first error-detecting/correcting protocol and a second error-detecting/correcting protocol different from the first error-detecting/correcting protocol;

a link establishment module fur establishing a communication link with the mobile station using the first error-detecting/correcting protocol over a transmission channel;

a quality of service monitor for determining a signal quality of the transmission channel;

a transmitter for sending a switchover message to the mobile station to switch from the first error-detecting/correcting protocol to the second error-detecting/correcting protocol if the signal quality passes the first predetermined threshold; and

a protocol selector for switching to the second error-detecting/correcting.

64. The base station of claim 63, wherein said transmitter sends a switchover message to the mobile station to switch from the second error-detecting/correcting protocol to the first error-detecting/correcting protocol if the signal quality passes a second predetermined threshold, and wherein the protocol selector switches to the first error-detecting/correcting protocol.

65. The base station of claim 63 further comprising a query module for querying the mobile station to determine if the mobile station supports the at least two service protocols, wherein the switchover message is transmitted only if the mobile station supports the at least two service protocols.

66. The base station of claim 63, wherein said switchover message includes a parameter indicating the service protocol.

67. The base station of claim 63, wherein said switchover message includes a parameter indicating a frame number.

68. The base station of claim 67, wherein said switching step occur after a frame having the frame number.

69. The base station of claim 63, wherein the first error-detecting/correcting protocol is less robust than the second error-detecting/correcting protocol, and wherein the transmitter sends the switchover message if the signal quality falls below the first predetermined threshold.

70. The base station of claim 63, wherein the first error-detecting/correcting protocol is more robust than the second error-detecting/correcting protocol, and wherein the transmitter sends the switchover message if the signal quality rises above the first predetermined threshold.

* * * * *

United States Patent [19]

Yamamoto et al.

[11] **Patent Number: 4,991,204**

[45] **Date of Patent: Feb. 5, 1991**

[54] **ADAPTIVE ROUTING CONTROL METHOD**

[75] Inventors: **Hisao Yamamoto; Kenichi Mase,** both of Tokyo; **Akiya Inoue,** Iruma; **Hiroo Itou,** Musashino; **Masato Suyama; Yoshitaka Hoshi,** both of Tokyo, all of Japan

[73] Assignee: **Nippon Telegraph and Telephone Corporation,** Tokyo, Japan

[21] Appl. No.: **433,949**

[22] Filed: **Nov. 9, 1989**

[30] **Foreign Application Priority Data**

Dec. 5, 1988 [JP] Japan 63-307474

Apr. 18, 1989 [JP] Japan 1-98297

[51] **Int. Cl.**[5] **H04Q 3/54; H04M 7/06;** H04M 3/36

[52] **U.S. Cl.** **379/221;** 379/113; 340/827

[58] **Field of Search** 379/113, 221, 230; 340/827

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,284,852 8/1981 Szybicki et al. 379/221

4,788,721 11/1988 Krishnan et al. .... 379/113 X

Primary Examiner—Stafford D. Schreyer

Attorney, Agent, or Firm—Pollock, Vande Sande and Priddy

[57] **ABSTRACT**

In a telecommunications network in which a plurality of switching nodes are interconnected via links each composed of a plurality of trunks and are each connected to a network control center via a control signal link, the network control center determines for each switching node a predetermined number of alternate routes for each first route on the basis of traffic data in the telecommunications network and supplies them as a set of available alternate routes to the switching node. The switching node assigns one or more of the available alternate routes in advance. The switching node responds to a call-connection request to try to connect the call to the first route, and when having failed in the call connection, the switching node retries the call connection through one of the assigned routes.

28 Claims, 15 Drawing Sheets

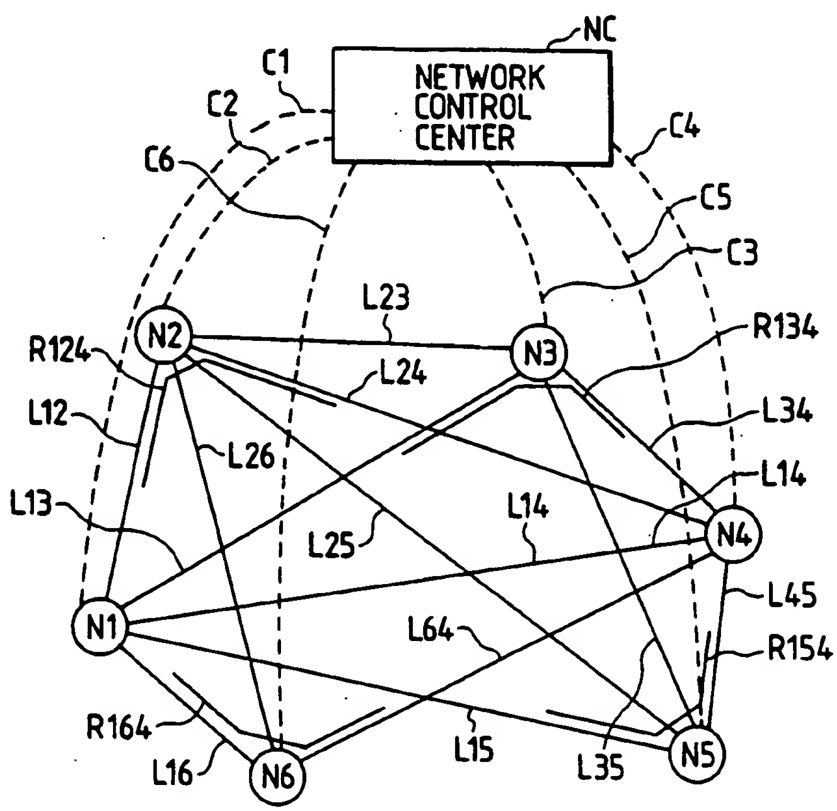


FIG. 1

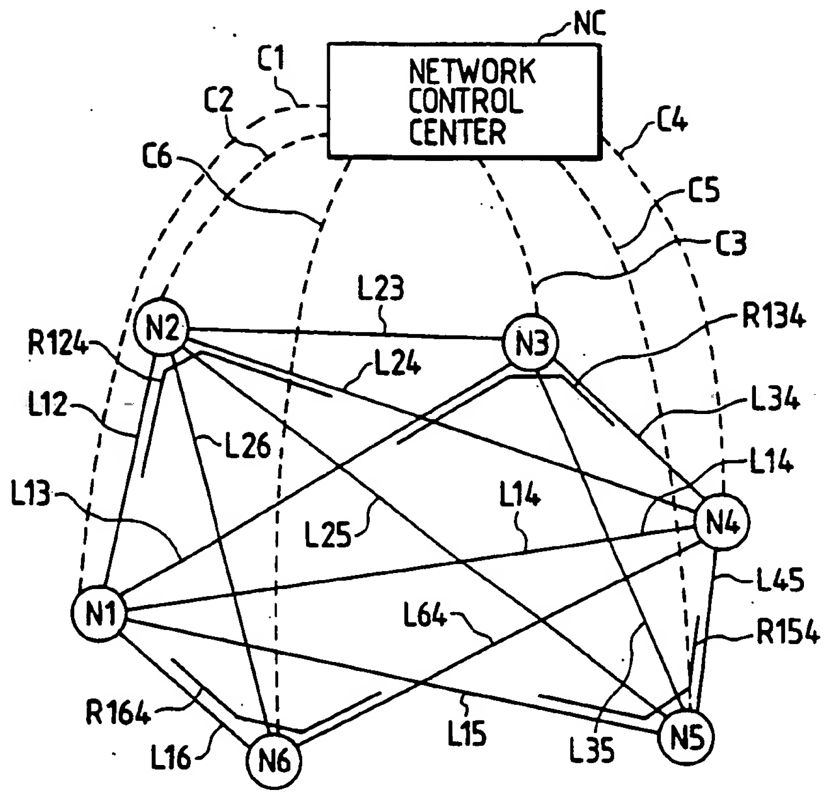


FIG. 3

FIRST ROUTE	SET OF AVAILABLE ALTERNATE ROUTES	CURRENTLY ASSIGNED ALTERNATE ROUTES
L12	R132, R142, R152	R142, R152
L13	R123, R143	R143
L14	R134, R154, R164	R134, R154, R164
L15	R145, R125	R125
L16	R126, R146	R126

FIG. 2

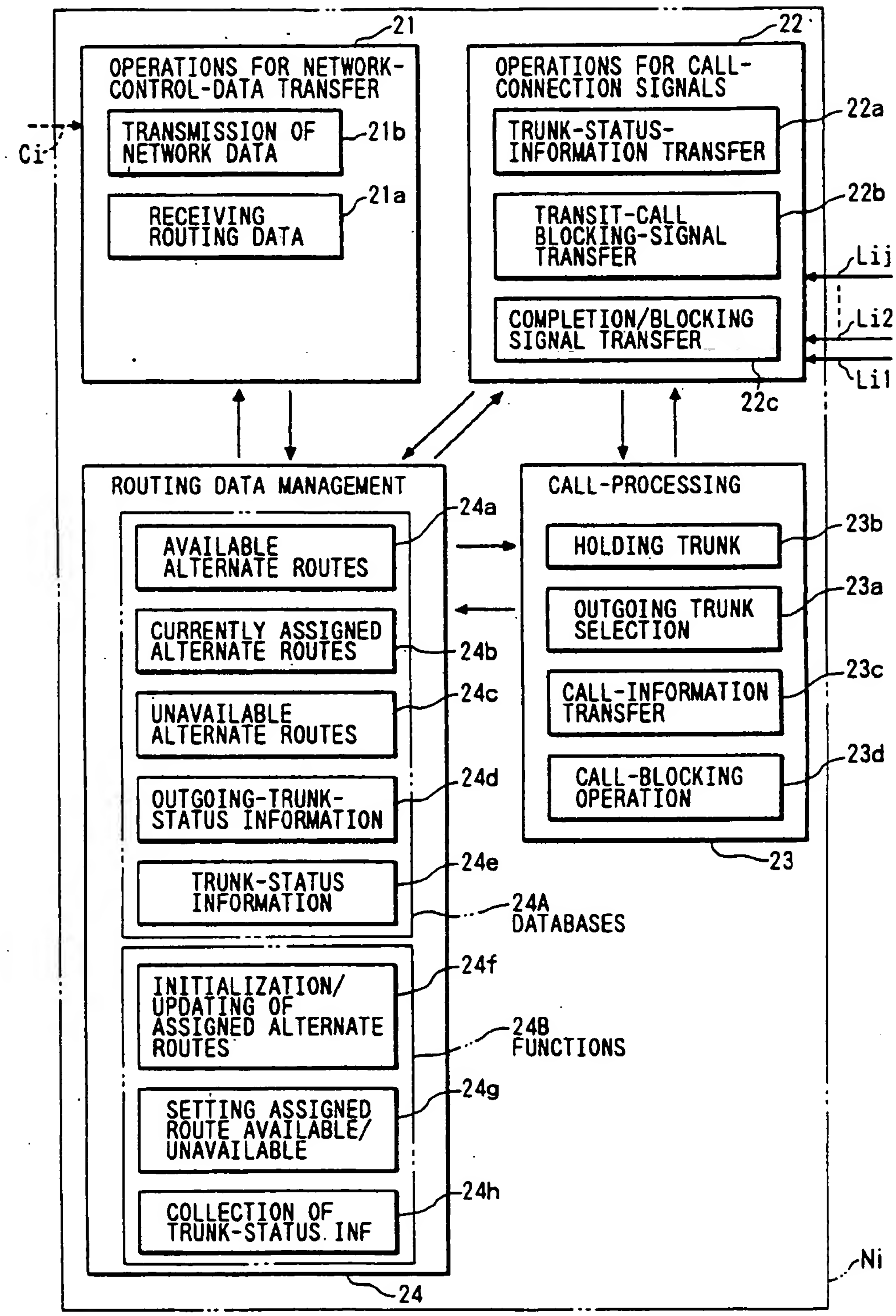

21
OPERATIONS FOR NETWORK-CONTROL-DATA TRANSFER
TRANSMISSION OF NETWORK DATA
21b
RECEIVING ROUTING DATA
21a
Ci
22
OPERATIONS FOR CALL-CONNECTION SIGNALS
TRUNK-STATUS-INFORMATION TRANSFER
22a
TRANSIT-CALL BLOCKING-SIGNAL TRANSFER
22b
COMPLETION/BLOCKING SIGNAL TRANSFER
22c
Lij
Li2
Li1
ROUTING DATA MANAGEMENT
AVAILABLE ALTERNATE ROUTES
24a
CURRENTLY ASSIGNED ALTERNATE ROUTES
24b
UNAVAILABLE ALTERNATE ROUTES
24c
OUTGOING-TRUNK-STATUS INFORMATION
24d
TRUNK-STATUS INFORMATION
24e
24A DATABASES
INITIALIZATION/ UPDATING OF ASSIGNED ALTERNATE ROUTES
24f
24B FUNCTIONS
SETTING ASSIGNED ROUTE AVAILABLE/ UNAVAILABLE
24g
COLLECTION OF TRUNK-STATUS. INF
24h
24
CALL-PROCESSING
HOLDING TRUNK
23b
OUTGOING TRUNK SELECTION
23a
CALL-INFORMATION TRANSFER
23c
CALL-BLOCKING OPERATION
23d
23
Ni

FIG. 4A

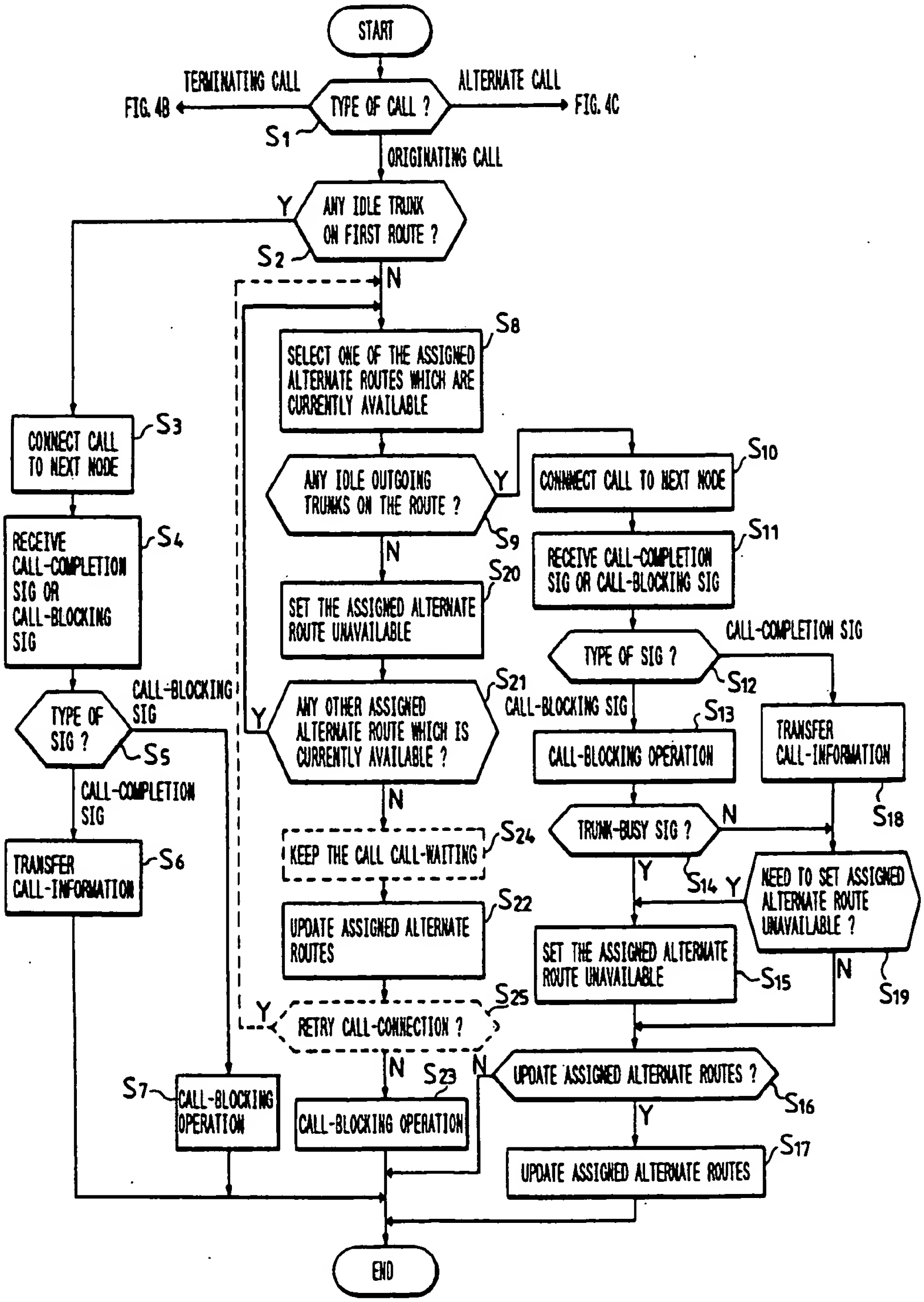

START
TYPE OF CALL ?
TERMINATING CALL
FIG. 4B
ALTERNATE CALL
FIG. 4C
S1
ORIGINATING CALL
ANY IDLE TRUNK ON FIRST ROUTE ?
S2
CONNECT CALL TO NEXT NODE
S3
RECEIVE CALL-COMPLETION SIG OR CALL-BLOCKING SIG
S4
TYPE OF SIG ?
S5
CALL-BLOCKING SIG
CALL-COMPLETION SIG
TRANSFER CALL-INFORMATION
S6
CALL-BLOCKING OPERATION
S7
SELECT ONE OF THE ASSIGNED ALTERNATE ROUTES WHICH ARE CURRENTLY AVAILABLE
S8
ANY IDLE OUTGOING TRUNKS ON THE ROUTE ?
S9
SET THE ASSIGNED ALTERNATE ROUTE UNAVAILABLE
S20
ANY OTHER ASSIGNED ALTERNATE ROUTE WHICH IS CURRENTLY AVAILABLE ?
S21
KEEP THE CALL CALL-WAITING
S24
UPDATE ASSIGNED ALTERNATE ROUTES
S22
RETRY CALL-CONNECTION ?
S25
CALL-BLOCKING OPERATION
S23
CONNECT CALL TO NEXT NODE
S10
RECEIVE CALL-COMPLETION SIG OR CALL-BLOCKING SIG
S11
TYPE OF SIG ?
S12
CALL-COMPLETION SIG
CALL-BLOCKING SIG
CALL-BLOCKING OPERATION
S13
TRANSFER CALL-INFORMATION
S18
TRUNK-BUSY SIG ?
S14
NEED TO SET ASSIGNED ALTERNATE ROUTE UNAVAILABLE ?
S19
SET THE ASSIGNED ALTERNATE ROUTE UNAVAILABLE
S15
UPDATE ASSIGNED ALTERNATE ROUTES ?
S16
UPDATE ASSIGNED ALTERNATE ROUTES
S17
END

FIG. 4B

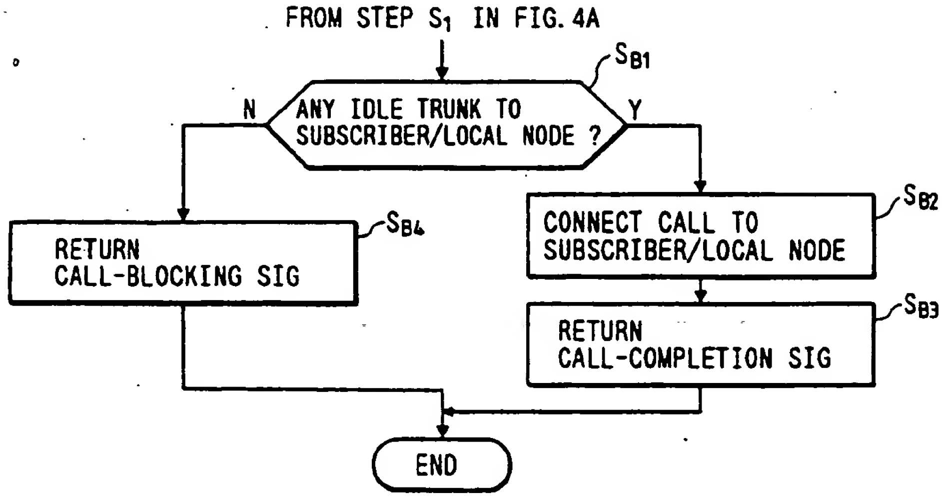

FROM STEP S1 IN FIG. 4A
SB1
ANY IDLE TRUNK TO SUBSCRIBER/LOCAL NODE ?
N
Y
SB2
CONNECT CALL TO SUBSCRIBER/LOCAL NODE
SB3
RETURN CALL-COMPLETION SIG
SB4
RETURN CALL-BLOCKING SIG
END

FIG. 4C

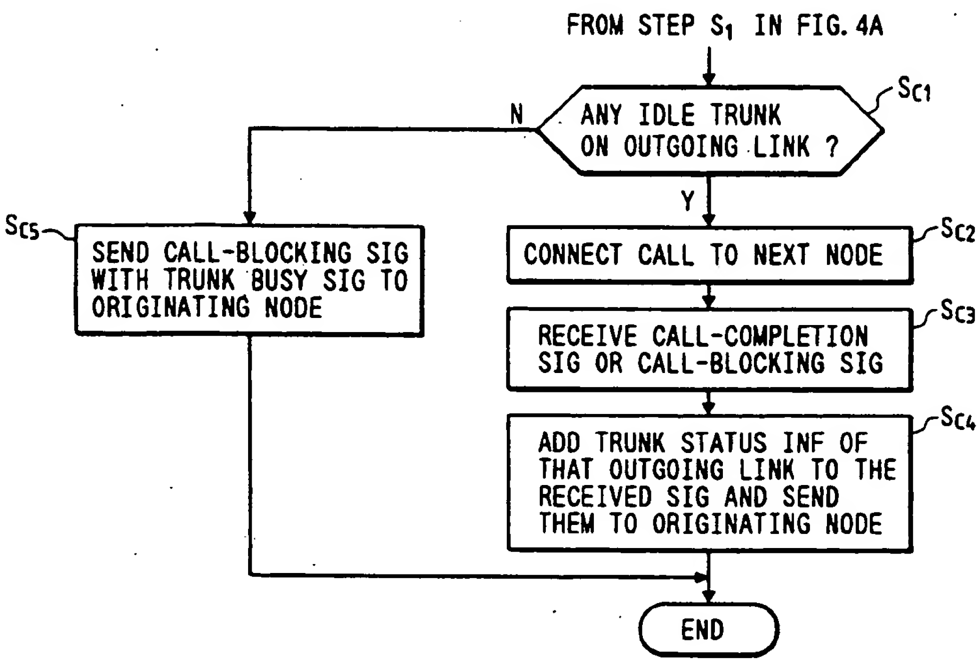

FROM STEP S1 IN FIG. 4A
SC1
ANY IDLE TRUNK ON OUTGOING LINK ?
N
Y
SC2
CONNECT CALL TO NEXT NODE
SC3
RECEIVE CALL-COMPLETION SIG OR CALL-BLOCKING SIG
SC4
ADD TRUNK STATUS INF OF THAT OUTGOING LINK TO THE RECEIVED SIG AND SEND THEM TO ORIGINATING NODE
SC5
SEND CALL-BLOCKING SIG WITH TRUNK BUSY SIG TO ORIGINATING NODE
END

FIG. 5

ASSIGNED ALTERNATE ROUTE	RELEASE TIME OF UNAVAILABLE STATUS
R134	20 : 10 : 30
R154	20 : 10 : 05
R164	00 : 00 : 00

FIG. 7

AVAILABLE ALTERNATE ROUTE	NUMBER OF IDLE TRUNKS	CHOICE PROBABILITY
R134	3	0.3
R154	5	0.5
R164	2	0.2

FIG. 6

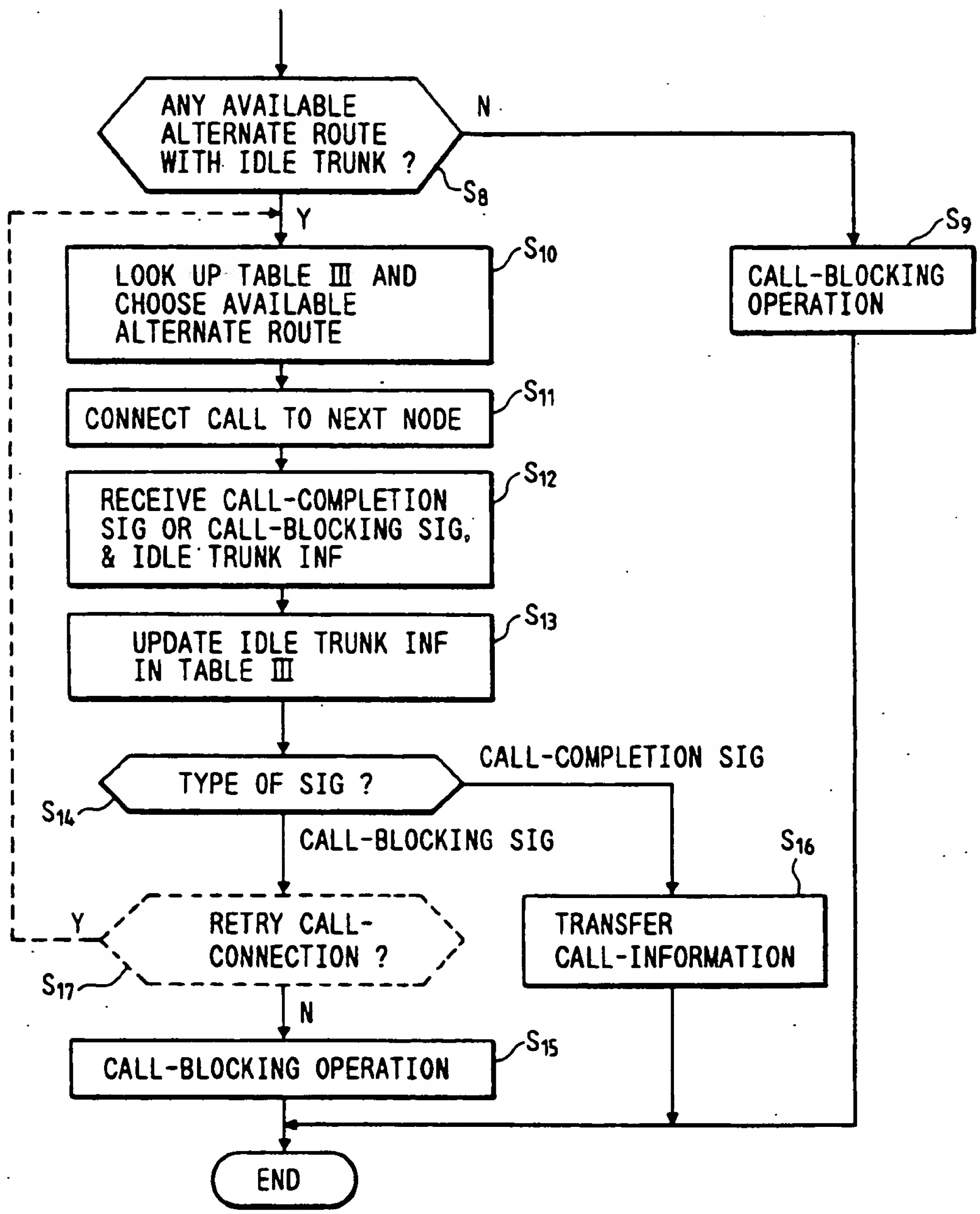

ANY AVAILABLE ALTERNATE ROUTE WITH IDLE TRUNK ?
N
S8
Y
LOOK UP TABLE III AND CHOOSE AVAILABLE ALTERNATE ROUTE
S10
S9
CALL-BLOCKING OPERATION
CONNECT CALL TO NEXT NODE
S11
RECEIVE CALL-COMPLETION SIG OR CALL-BLOCKING SIG, & IDLE TRUNK INF
S12
UPDATE IDLE TRUNK INF IN TABLE III
S13
TYPE OF SIG ?
CALL-COMPLETION SIG
S14
CALL-BLOCKING SIG
S16
RETRY CALL-CONNECTION ?
Y
S17
N
TRANSFER CALL-INFORMATION
CALL-BLOCKING OPERATION
S15
END

FIG. 8

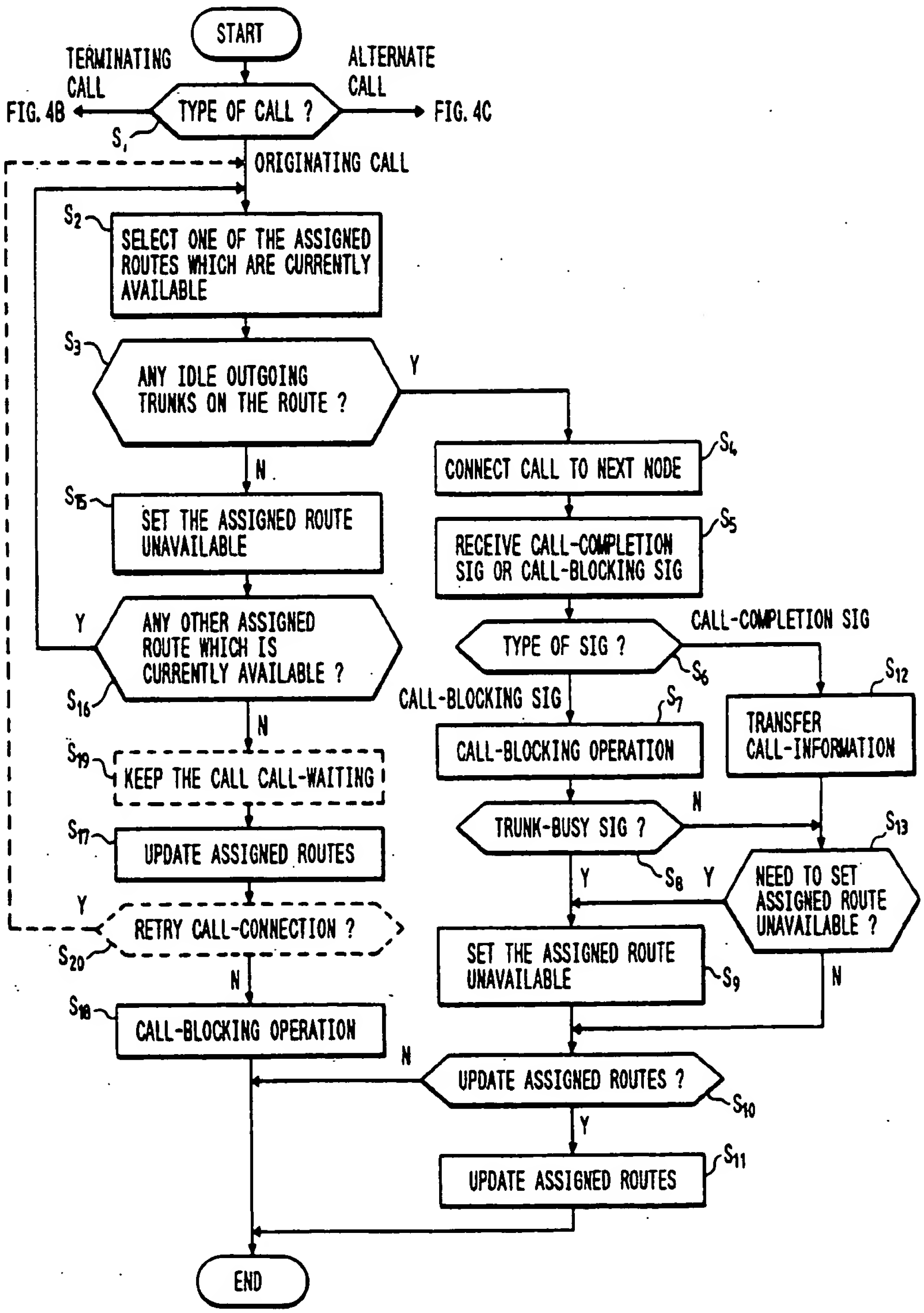

START
TERMINATING CALL
ALTERNATE CALL
FIG. 4B
TYPE OF CALL ?
FIG. 4C
S1
ORIGINATING CALL
S2
SELECT ONE OF THE ASSIGNED ROUTES WHICH ARE CURRENTLY AVAILABLE
S3
ANY IDLE OUTGOING TRUNKS ON THE ROUTE ?
Y
N
CONNECT CALL TO NEXT NODE
S4
S15
SET THE ASSIGNED ROUTE UNAVAILABLE
RECEIVE CALL-COMPLETION SIG OR CALL-BLOCKING SIG
S5
Y
ANY OTHER ASSIGNED ROUTE WHICH IS CURRENTLY AVAILABLE ?
S16
TYPE OF SIG ?
S6
CALL-COMPLETION SIG
CALL-BLOCKING SIG
S7
S12
N
CALL-BLOCKING OPERATION
TRANSFER CALL-INFORMATION
S19
KEEP THE CALL CALL-WAITING
N
TRUNK-BUSY SIG ?
S17
UPDATE ASSIGNED ROUTES
S13
Y
S8
Y
NEED TO SET ASSIGNED ROUTE UNAVAILABLE ?
Y
RETRY CALL-CONNECTION ?
S20
SET THE ASSIGNED ROUTE UNAVAILABLE
S9
N
N
S18
CALL-BLOCKING OPERATION
N
UPDATE ASSIGNED ROUTES ?
S10
Y
S11
UPDATE ASSIGNED ROUTES
END

FIG. 9

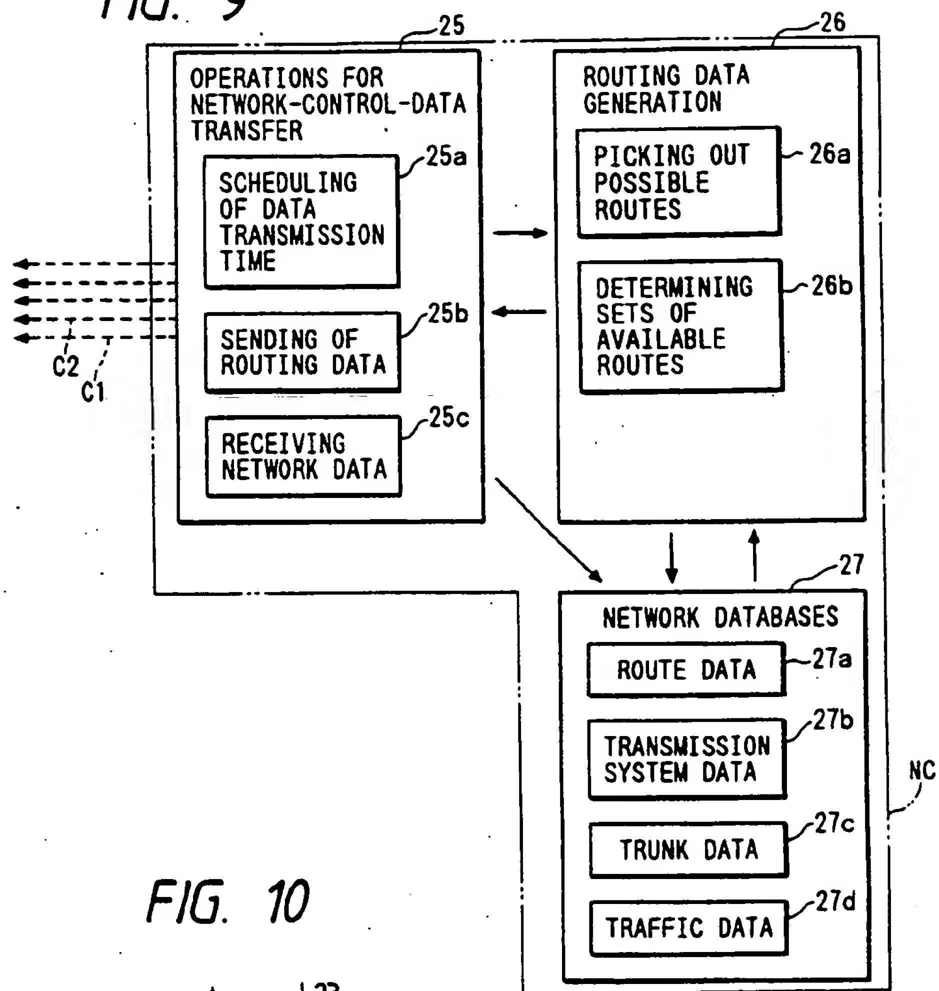

25
OPERATIONS FOR NETWORK-CONTROL-DATA TRANSFER
25a
SCHEDULING OF DATA TRANSMISSION TIME
25b
SENDING OF ROUTING DATA
25c
RECEIVING NETWORK DATA
C2
C1
26
ROUTING DATA GENERATION
26a
PICKING OUT POSSIBLE ROUTES
26b
DETERMINING SETS OF AVAILABLE ROUTES
27
NETWORK DATABASES
27a
ROUTE DATA
27b
TRANSMISSION SYSTEM DATA
27c
TRUNK DATA
27d
TRAFFIC DATA
NC

FIG. 10

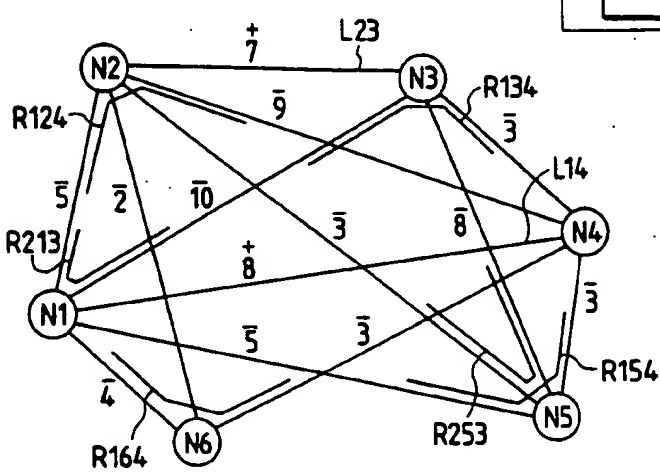

N1
N2
N3
N4
N5
N6
L23
L14
R124
R213
R134
R154
R164
R253

FIG. 11

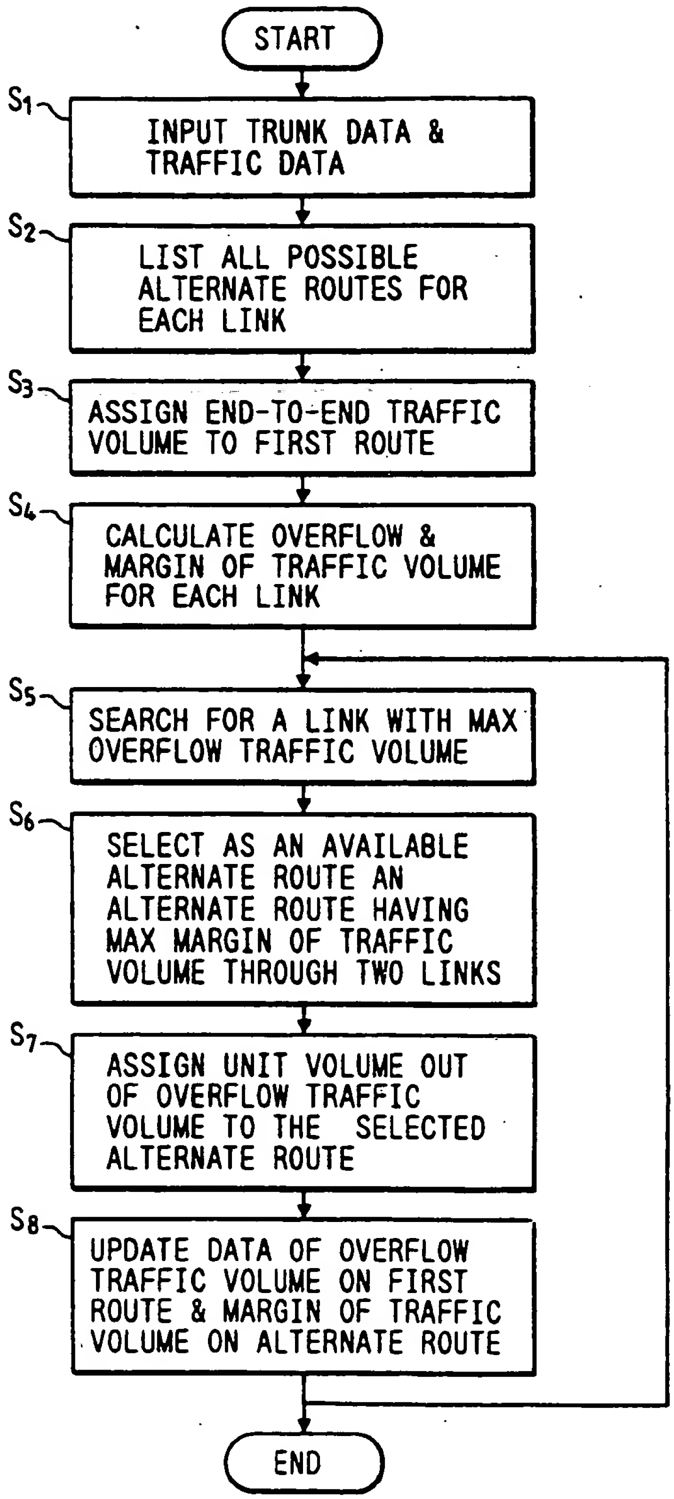

START
S1
INPUT TRUNK DATA & TRAFFIC DATA
S2
LIST ALL POSSIBLE ALTERNATE ROUTES FOR EACH LINK
S3
ASSIGN END-TO-END TRAFFIC VOLUME TO FIRST ROUTE
S4
CALCULATE OVERFLOW & MARGIN OF TRAFFIC VOLUME FOR EACH LINK
S5
SEARCH FOR A LINK WITH MAX OVERFLOW TRAFFIC VOLUME
S6
SELECT AS AN AVAILABLE ALTERNATE ROUTE AN ALTERNATE ROUTE HAVING MAX MARGIN OF TRAFFIC VOLUME THROUGH TWO LINKS
S7
ASSIGN UNIT VOLUME OUT OF OVERFLOW TRAFFIC VOLUME TO THE SELECTED ALTERNATE ROUTE
S8
UPDATE DATA OF OVERFLOW TRAFFIC VOLUME ON FIRST ROUTE & MARGIN OF TRAFFIC VOLUME ON ALTERNATE ROUTE
END

FIG. 12A

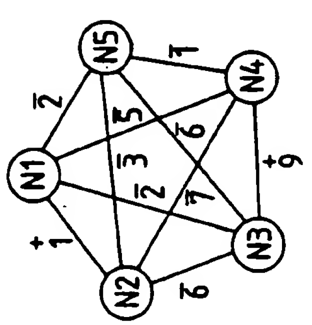


FIG. 12B

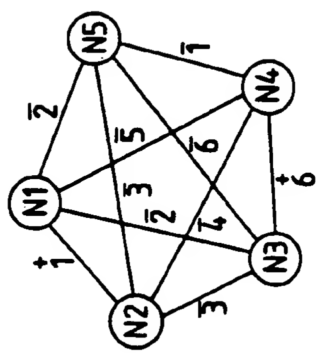


FIG. 12C

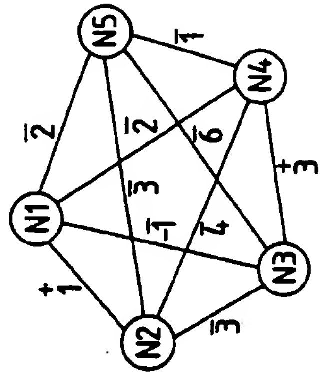


FIG. 12D

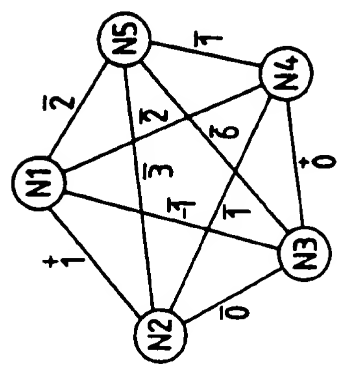


FIG. 12E

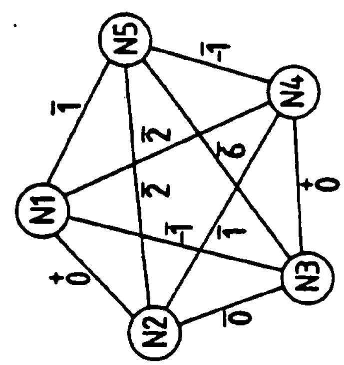


FIG. 12F

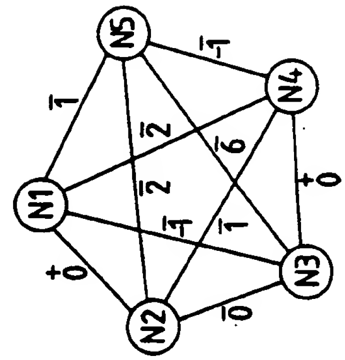


FIG. 13

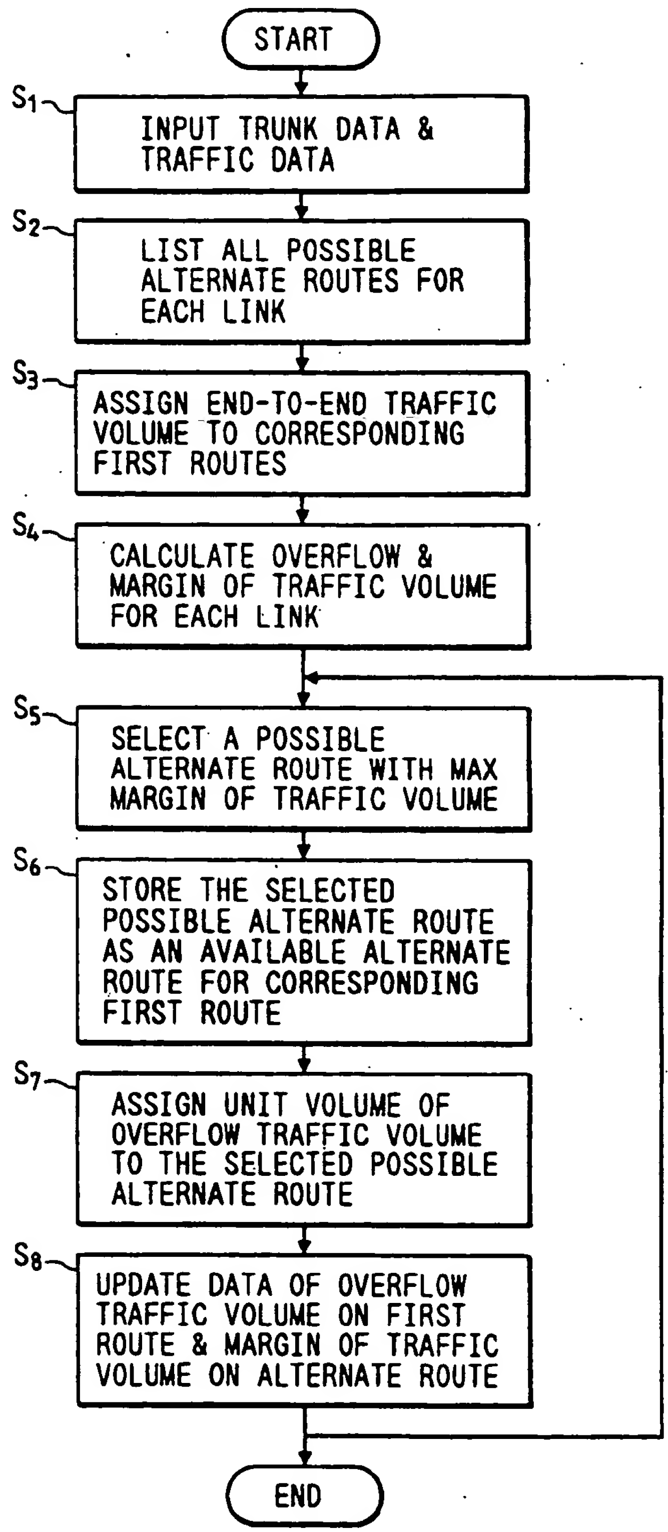

START
S1
INPUT TRUNK DATA & TRAFFIC DATA
S2
LIST ALL POSSIBLE ALTERNATE ROUTES FOR EACH LINK
S3
ASSIGN END-TO-END TRAFFIC VOLUME TO CORRESPONDING FIRST ROUTES
S4
CALCULATE OVERFLOW & MARGIN OF TRAFFIC VOLUME FOR EACH LINK
S5
SELECT A POSSIBLE ALTERNATE ROUTE WITH MAX MARGIN OF TRAFFIC VOLUME
S6
STORE THE SELECTED POSSIBLE ALTERNATE ROUTE AS AN AVAILABLE ALTERNATE ROUTE FOR CORRESPONDING FIRST ROUTE
S7
ASSIGN UNIT VOLUME OF OVERFLOW TRAFFIC VOLUME TO THE SELECTED POSSIBLE ALTERNATE ROUTE
S8
UPDATE DATA OF OVERFLOW TRAFFIC VOLUME ON FIRST ROUTE & MARGIN OF TRAFFIC VOLUME ON ALTERNATE ROUTE
END

FIG. 14

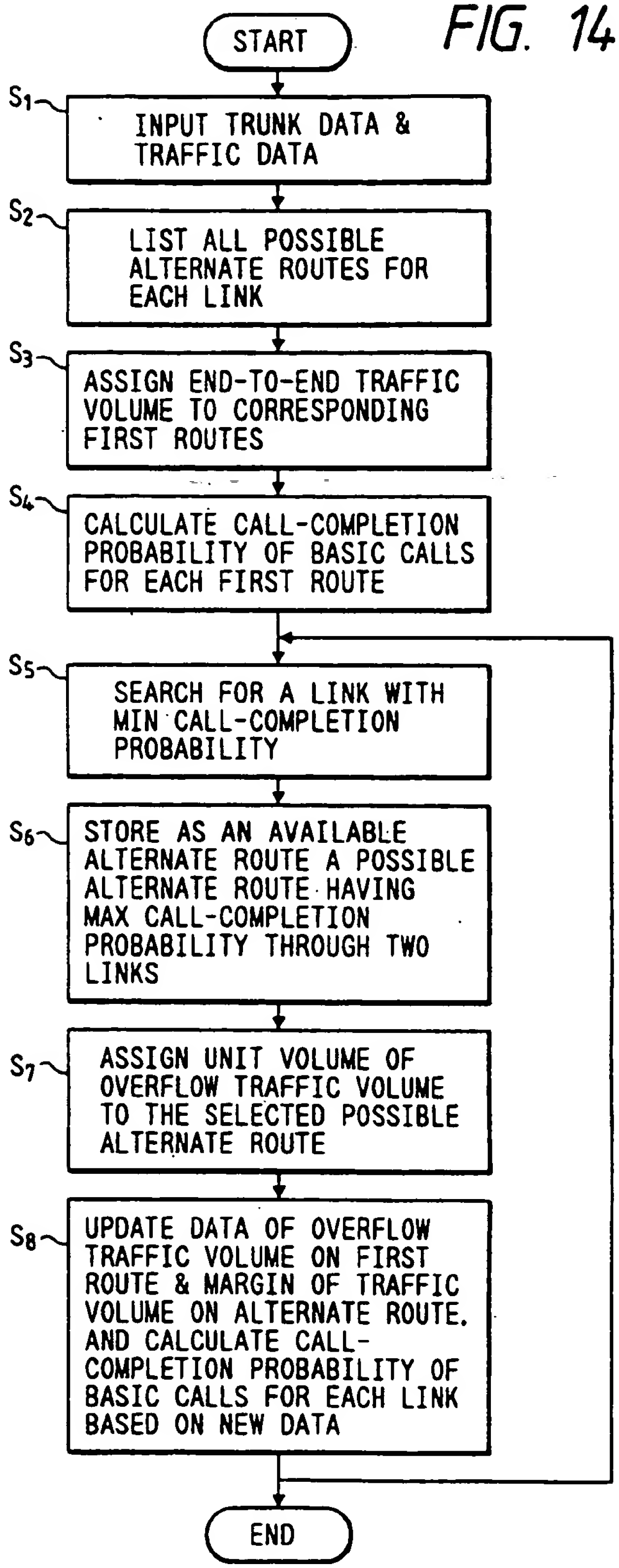

START
S1
INPUT TRUNK DATA & TRAFFIC DATA
S2
LIST ALL POSSIBLE ALTERNATE ROUTES FOR EACH LINK
S3
ASSIGN END-TO-END TRAFFIC VOLUME TO CORRESPONDING FIRST ROUTES
S4
CALCULATE CALL-COMPLETION PROBABILITY OF BASIC CALLS FOR EACH FIRST ROUTE
S5
SEARCH FOR A LINK WITH MIN CALL-COMPLETION PROBABILITY
S6
STORE AS AN AVAILABLE ALTERNATE ROUTE A POSSIBLE ALTERNATE ROUTE HAVING MAX CALL-COMPLETION PROBABILITY THROUGH TWO LINKS
S7
ASSIGN UNIT VOLUME OF OVERFLOW TRAFFIC VOLUME TO THE SELECTED POSSIBLE ALTERNATE ROUTE
S8
UPDATE DATA OF OVERFLOW TRAFFIC VOLUME ON FIRST ROUTE & MARGIN OF TRAFFIC VOLUME ON ALTERNATE ROUTE. AND CALCULATE CALL-COMPLETION PROBABILITY OF BASIC CALLS FOR EACH LINK BASED ON NEW DATA
END

FIG. 15

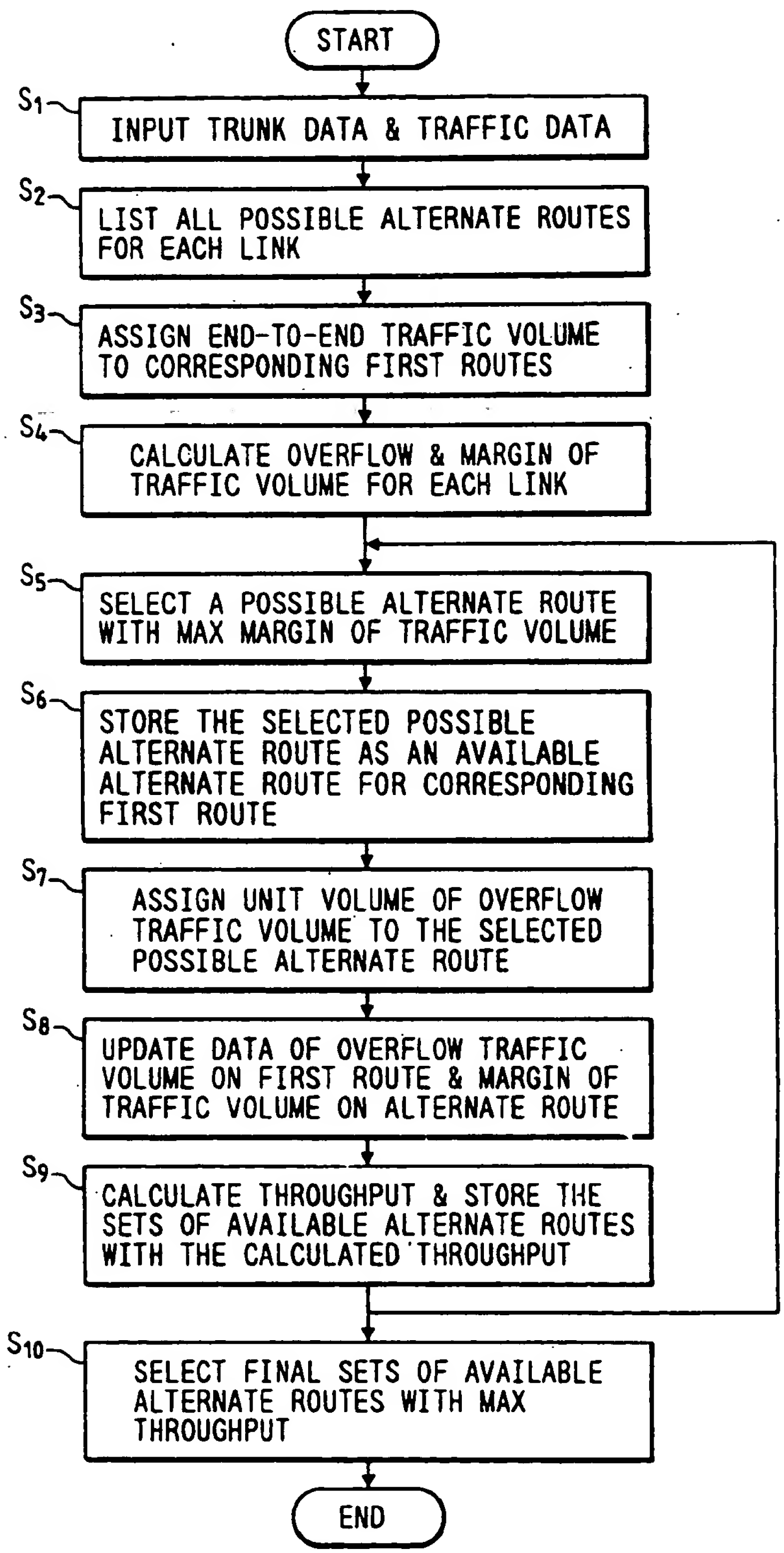

START
S1
INPUT TRUNK DATA & TRAFFIC DATA
S2
LIST ALL POSSIBLE ALTERNATE ROUTES FOR EACH LINK
S3
ASSIGN END-TO-END TRAFFIC VOLUME TO CORRESPONDING FIRST ROUTES
S4
CALCULATE OVERFLOW & MARGIN OF TRAFFIC VOLUME FOR EACH LINK
S5
SELECT A POSSIBLE ALTERNATE ROUTE WITH MAX MARGIN OF TRAFFIC VOLUME
S6
STORE THE SELECTED POSSIBLE ALTERNATE ROUTE AS AN AVAILABLE ALTERNATE ROUTE FOR CORRESPONDING FIRST ROUTE
S7
ASSIGN UNIT VOLUME OF OVERFLOW TRAFFIC VOLUME TO THE SELECTED POSSIBLE ALTERNATE ROUTE
S8
UPDATE DATA OF OVERFLOW TRAFFIC VOLUME ON FIRST ROUTE & MARGIN OF TRAFFIC VOLUME ON ALTERNATE ROUTE
S9
CALCULATE THROUGHPUT & STORE THE SETS OF AVAILABLE ALTERNATE ROUTES WITH THE CALCULATED THROUGHPUT
S10
SELECT FINAL SETS OF AVAILABLE ALTERNATE ROUTES WITH MAX THROUGHPUT
END

FIG. 16

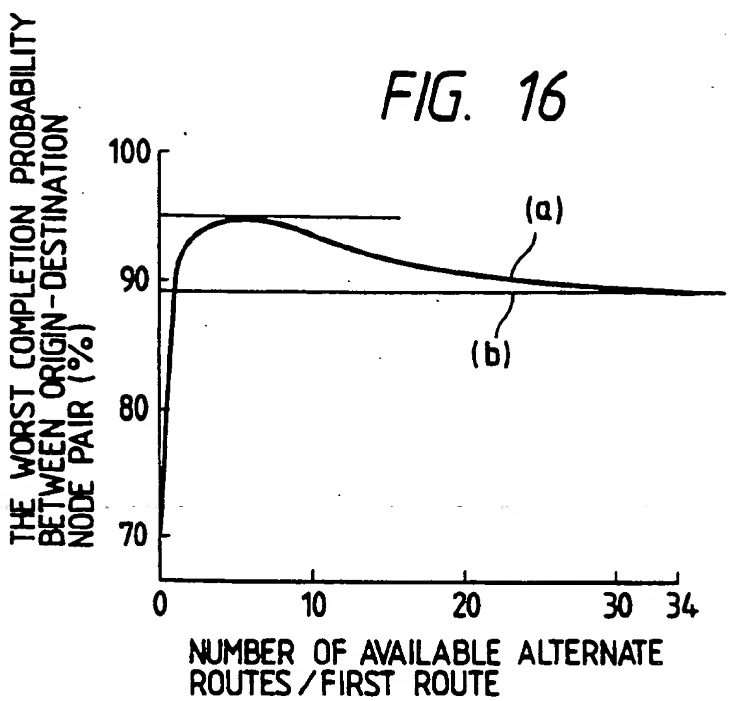
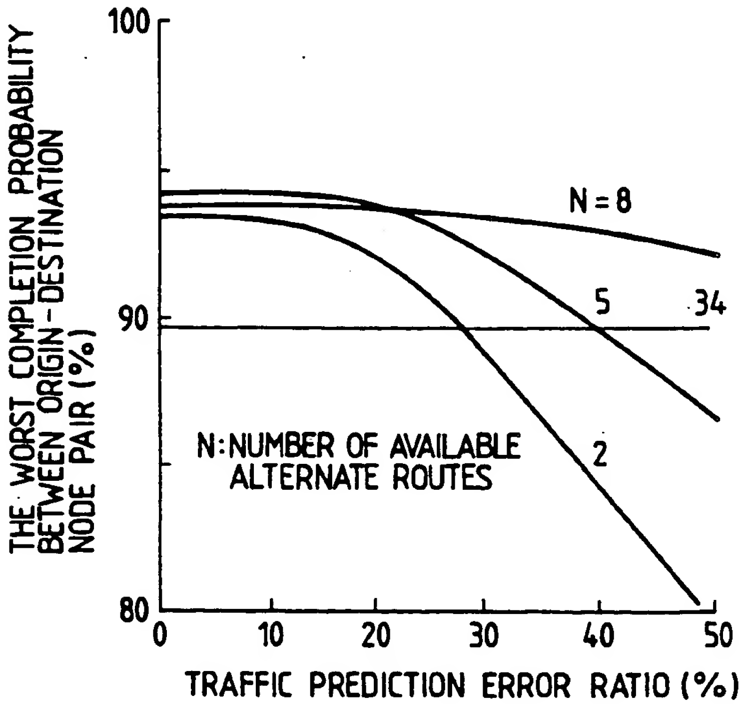

THE WORST COMPLETION PROBABILITY BETWEEN ORIGIN-DESTINATION NODE PAIR (%)
100
90
80
70
(a)
(b)
0
10
20
30
34
NUMBER OF AVAILABLE ALTERNATE ROUTES / FIRST ROUTE

FIG. 17


THE WORST COMPLETION PROBABILITY BETWEEN ORIGIN-DESTINATION NODE PAIR (%)
100
90
80
N = 8
5
34
2
N: NUMBER OF AVAILABLE ALTERNATE ROUTES
0
10
20
30
40
50
TRAFFIC PREDICTION ERROR RATIO (%)

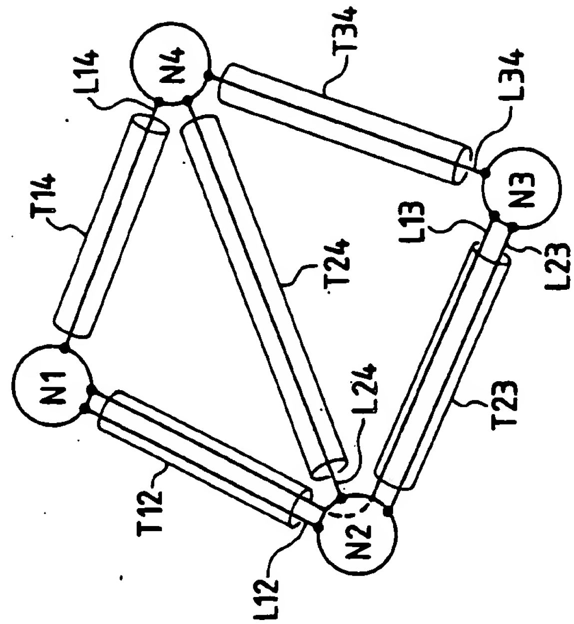

N1
N2
N3
N4
T12
T14
T23
T24
T34
L12
L13
L14
L23
L24
L34

FIG. 19

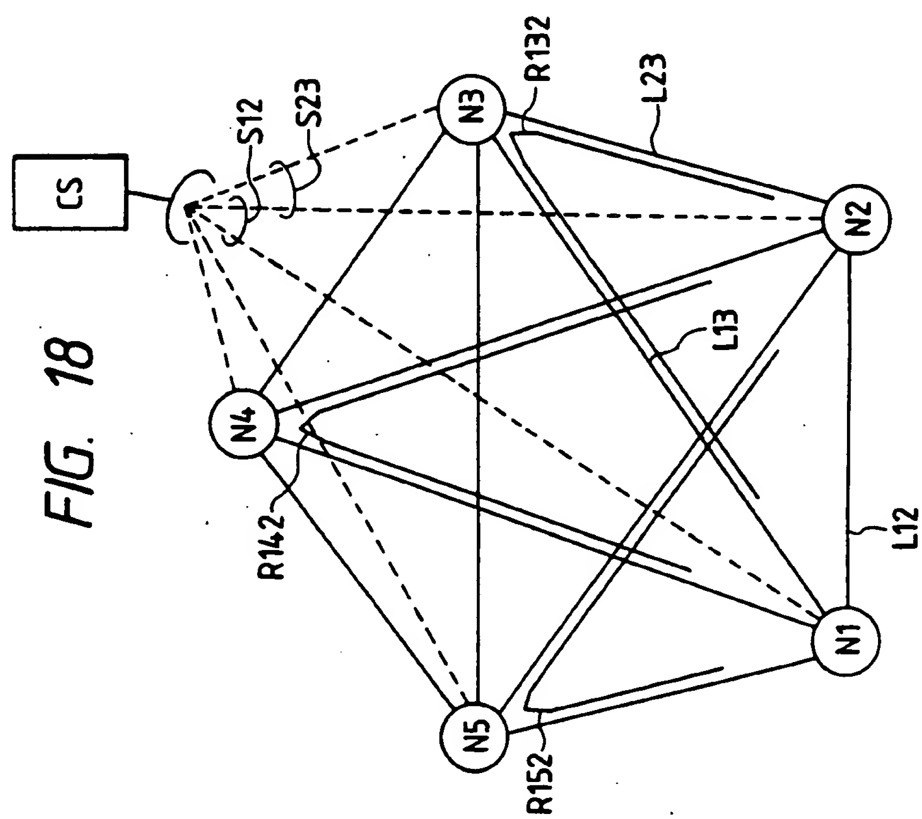

CS
S12
S23
N1
N2
N3
N4
N5
R132
R142
R152
L12
L13
L23

FIG. 18

ADAPTIVE ROUTING CONTROL METHOD

BACKGROUND OF THE INVENTION

The present invention relates to a route selection method for telecommunications networks and, more particularly, to an adaptive routing control method which permits optimum routing according to the network status (trunk usage, offered traffic volume, or congestion conditions).

In telecommunications networks with a plurality of switching nodes routes for interconnecting them usually include a first route which achieves the most economical call connection between each originating-terminating node pair. When the first route is not busy, the first route is used to interconnect the originating and terminating nodes, whereas when the first route is busy, alternates routes can be established via one or more other switching nodes. With such a conventional route selection algorithm, however, switching nodes through which alternate routes can be established are limited and the order of their selection also is fixed because of technical restrictions inherent to the call-connection control system employed.

With the recent introduction of switching nodes of a stored program control system and a common channel signaling inter-office system for an inter-office signal transfer, it has become possible to utilize, in place of the above-mentioned route selection algorithm, a dynamic routing method which affords flexible routing based on the distribution of idle trunks in the network.

The dynamic routing method may be classified into time-dependent routing and state-dependent routing (see B. R. Hurley, et al., "A Survey of Dynamic Routing Methods for Circuit Switched Traffic," IEEE COMMUNICATIONS MAGAZINE, Vol. 25, No. 9, pp. 13–21, September 1987, for example).

The time-dependent routing is a method in which a suitable routing pattern is preset for each predetermined time slot, i.e. a method in which a set of alternate routes and the order of their selection are preset for each first route and a call originating in a switching node is connected to the intended destination node, following the routing pattern preset for the time slot concerned. A typical example of the time-dependent routing is a DNHR (Dynamic Nonhierarchical Routing) system proposed by AT & T, Inc. of the United States (see G. R. Ash, et al., "Design and Optimization of Networks with Dynamic Routing," BSTJ, Vol. 60, pp. 1787–1820, October 1981, for instance).

The state-dependent routing is a method which performs a call connection while updating the routing pattern in real time in accordance with the network status such as trunk usage in the network. This method is implemented by centralized or distributed control.

In the state-dependent routing method by centralized control a network control center collects data about the trunk usage throughout the network, calculates a routing pattern between each originating-terminating node pair, and indicates the routing pattern to each switching node in real time. An example of this state-dependent routing method by centralized control is a TSMR system proposed by AT & T, Inc. of the United States and a DCR system by Northern Telecom of Canada (see the afore-mentioned literature by B. R. Hurley, et al., for instance).

In the state-dependent routing method by distributed control each switching node independently detects the network status and autonomously searches for an alternate route based on the network status information, thereby setting an appropriate routing pattern between an origin-destination node pair. Examples of this method are those proposed by British Telecommunications of Great Britain and Centre National D'etudes des Telecommunications of France (commonly known as "CENT"). Both methods are common in basic principle, and the method by British Telecommunications is called a DAR system (see B. R. Stacey, et al., "Dynamic Alternative Routing in the British Telecom Trunk Network," International Switching Symposium, ISS-87, B12.4.1–B.12.4.5, 1987, or Hennion B., "Feedback Methods for Calls Allocation on the Crossed Traffic Routing," International Teletraffic Congress, ITC-9, pp. HEENNION-1 to HENNION-3, 1979, for example).

Some proposals have been made so far for the dynamic routing as mentioned above but they have the following problems yet to be solved for practical use.

(i) The time-dependent routing of the aforementioned DNHR system, for instance, would work well in a country like the United States where a plurality of standard times are used, the traffic busy hour differs sharply with regions, an appropriate routing pattern for each time slot can be forecast, and updating of the routing pattern can be scheduled. Where the traffic busy hour is common almost all over the country as in Japan, however, the time-dependent routing, if used singly, would not be so effective. In a country like Japan it is of prime importance to efficiently handle offered traffic, quickly responding to an excess or shortage of the trunk-number of transit links which is caused by restrictions on the management of trunk resources such as the trunk assignment interval, the trunk modularity, etc. or unpredictable traffic variations, and the state-dependent routing is more effective rather than the time-dependent routing.

(ii) In general, the state-dependent routing by centralized control permits efficient routing, because a routing pattern can be indicated based on the optimization of the entire network through observation of its status, for example, the trunk usage in the network. However, in the case where the observation cycle is long or an information transfer delay occurs, that is, where a time lag is great between the observation and the execution of a call connection by a routing pattern based on the observation, the state of the network varies during this time resulting in an increase in the probability of effecting erroneous control. This will not produce the intended effect and will lower the call-connection quality.

To avoid such a problem and hence achieve the intended effect, it is necessary to reduce the network status observation cycle and the switching node control cycle. The aforementioned TSMR or DCR system, for example, premises that both cycles are within 10 seconds. In a large-scale telecommunications network in which the number of switching nodes to be controlled is several hundreds and the number of links to be measured is as large as tens of thousands, however, such a high-speed observation and control are difficult. In other words, the amount of data to be processed by the network control center, the amount of data to be transferred between the switching nodes and the network control center, and measurements in the switching nodes and the amount of data to be transmitted and received among them are enormous and the facilities therefor are also vast, resulting in an uneconomical

system. In addition, a failure in the control center of such a large-scale network will throw the network into disorder.

(iii) With the a aforementioned DAR system and the self-routing system in the state-dependent routing by distributed control, no network control center is employed and each switching node checks the status of alternate routes by a signal handled in its call-connection procedure and autonomously changes an alternate route accordingly, thereby implementing a preferably routing pattern throughout the telecommunications network. Consequently, the problem mentioned above in (ii) can be avoided. In a large-scale telecommunications network, however, the number of alternate routes for each origin-destination node pair becomes appreciable, incurring various disadvantages. For instance, in a telecommunications network which forms a mesh by 100 switching nodes the number of alternate routes via two transit links between each origin-destination node pair alone is as large as 98.

In such an instance, (a) alternate routes are rechecked through a search by trial and error prior to a call-connection procedure, and consequently, when the number of available alternate routes is unnecessarily large, the search is repeated inevitably many times until a routing pattern updated according to temporary traffic variations is restored to its initial state. Similarly, when a traffic pattern throughout the network changes or transmission equipment breaks down, the search is repeated many times until each switching node shifts to a new favorable routing pattern. This will deteriorate the call-connection quality and increase the amount of data to be processed by each switching node. (b) An increase in the amount of data managed by each switching node calls for an increase in the number of tables for processing data and the number of counters for counting the number of calls. That is to say, the amount of data which is managed for each origin-destination node pair or each first route increases, and consequently, alternate route tables are required and the state of alternate routing must be monitored from the viewpoint of network management. This necessitates a number of counters for counting the number and the traffic volume of alternate calls and the transit-call-completion probability in each alternate route. Moreover, (c) an increase in the number of counters used will cause an increase in the computer running time to be processed for measurement by the counters.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide an adaptive routing control method which obviates the above-mentioned defects of the prior art, enables an optimum alternate route to be selected in accordance with real time traffic variations and the current network conditions (which consist of a network topology and a matrix of the number of trunks between each node pair), and affords the reduction of the amount of data to be managed by each switching node and the number of tables and counters used even in a large-scale telecommunications network.

To attain the above objective, in the telecommunications network to which the adaptive routing control method of the present invention is applied, a plurality of switching nodes are interconnected via links each composed of a plurality of trunks, one or more routes each composed of a set of one or more links are present between each node pair, and at least one network control center is connected via a control signal link to each switching node. According to the present invention, the network control center adaptively determines, for each node pair, a set of available routes each composed of one or more routes which are set available in accordance with the traffic volume in the telecommunications network and the number of trunks set for each link. The network control center sends the sets of available routes to each switching node and, at a predetermined time, updates the set of available routes and resends them to each switching node. Each switching node responds to a call-connection request to select one of the available routes and performs a required call-connection procedure.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram illustrating an example of the telecommunications network to which the adaptive routing control method of the present invention is applied;

FIG. 2 is a function-block-chart of a network control center NC in the telecommunications network depicted in FIG. 1;

FIG. 3 is a table I showing sets of available alternate routes for respective first routes and currently assigned routes, provided to a switching node N1 in the telecommunications network depicted in FIG. 1;

FIG. 4A is a flowchart showing a call-connection procedure in an originating node;

FIG. 4B is a flowchart showing a call-connection procedure in a terminating node;

FIG. 4C is a flowchart showing a call-connection procedure in a transit node;

FIG. 5 is a table II showing available or unavailable status of assigned alternate routes;

FIG. 6 is a flowchart showing another example of the call-connection procedure in the originating node;

FIG. 7 is a table III showing the numbers of idle trunks recorded for respective alternate routes and their choice probabilities determined in accordance with them;

FIG. 8 is a flowchart showing another example of the call-connection procedure in the originating node according to the routing control method of the present invention;

FIG. 9 is a function-block-chart of a network control center of the telecommunications network;

FIG. 10 is a schematic diagram showing an overflow traffic volume or the margin of traffic volume calculated for each link on the basis of the end-to-end traffic volume in the telecommunications network so as to determine a set of available alternate routes for each link;

FIG. 11 is a flowchart showing an example of the procedure for determining the sets of available alternate routes;

FIGS. 12A through 12F are schematic diagrams showing an example of the procedure for determining the sets of available alternate routes;

FIG. 13 is a flowchart showing another example of the procedure for determining the sets of available alternate routes;

FIG. 14 is a flowchart showing another example of the procedure for determining the sets of available alternate routes;

FIG. 15 is a flowchart showing still another example of the procedure for determining the sets of available alternate routes;

FIG. 16 is a graph showing the number of available alternate routes in each set and the call-completion probability, for explaining the effect of the present invention;

FIG. 17 is a graph showing the relationship between calculated traffic forecasting errors and the call-completion probability, for explaining the effect of the present invention;

FIG. 18 is a schematic diagram showing a telecommunications network including a communications satellite link to which the routing control method of the present invention can be applied; and

FIG. 19 is a schematic diagram for explaining the relationship between a transmission network and communication links in the telecommunications network.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In FIG. 1 there is shown the general constitution of the telecommunications network embodying the adaptive routing control method of the present invention. A plurality of switching nodes N1 to N6 are interconnected via solid-line links L12, L23, . . . to form various routes. The links L12, L23, . . . each include a preset number of trunks. A network control center NC is provided in association with these switching nodes N1 to N6. The switching nodes N1 to N6 are connected to the network control center NC via broken-line control signal links C1 to C6, respectively. The switching nodes N1 to N6 each possess the functions of an originating node which originates a call, a transit node which relays the call, and a terminating node which is the destination of the call.

Now, definitions will be given of some terms which will be used in the following description of embodiments of the present invention.

First Route: A predetermined route which connects two arbitrary switching nodes for a call-connection. When there is one link which directly connects the two switching nodes, it is used as the first route, and when such a link is not found, a predetermined route is used as the first route which connects them via one or more other switching nodes (i.e. transit nodes).

Possible Routes: Routes through which two arbitrary switching nodes can be connected in the communications network shown in FIG. 1. In the description of the present invention they are defined as routes, each formed by a maximum of two links.

Set of Available Routes: One or more routes selected by the network control center from all the possible routes.

Alternate Routes: Possible routes except the first route.

Outgoing Link: A link from which a call is sent out from each switching node.

First and Second Alternate Links: A link which connects an originating and a transit node over an alternate route will be referred to as a first alternate link. A link which connects the transit node and a terminating node will be referred to as a second link.

Set of Available Alternate Routes: One or more alternate routes preselected by the network control center from all alternate routes for the first route which connects two arbitrary switching nodes.

In the embodiment of the present invention which is applied to the telecommunications network depicted in FIG. 1 the network control center NC predetermines, for each time slot, sets of available alternate routes which are used by the switching nodes N1 to N6, respectively, and transfers the predetermined sets of available alternate routes to the switching nodes N1 to N6 at predetermined times. The switching nodes N1 to N6 each respond to a call-connection request to preferentially search the first route for an idle trunk, and when no idle trunk is found in the first route, the switching node tries a call-connection via an alternate route adaptively selected, in accordance with the trunk usage, from the set of available alternate routes transferred from the network control center NC. In the following description a link which directly connects two arbitrary switching nodes Ni and Nj (where i and j are positive integers, which are not equal to each other) will be identified by Lij and a route which connects the two switching nodes via transit node Nk (where k is a positive integer, which is not equal to the integers i and j) will be identified by Rikj.

Switching Node

FIG. 2 is a function-block-chart of each of the switching nodes N1 to N6 in the telecommunications network shown in FIG. 1. The switching node Ni performs network-control-data transfer operations 21, call-connection signal processing 22, call processing 23 and routing data management 24. The network-control-data transfer operations 21 includes an operation 21*a* of receiving routing data, i.e. sets of available alternate routes from the network control center NC via the control signal link Ci and an operation 21*b* of transmitting network data to the network control center NC via the control signal link Ci. The call-connection signal processing 22 includes: a trunk-status-information transfer operation 22*a* of receiving trunk status information from other switching nodes or transmitting trunk status information in the switching node Ni via the links Li1, Li2, . . . , Lij, . . . ; a transit-call-blocking signal transfer operation 22*b* of sending a transit-call-blocking signal back to an originating node in the case of a failure in the transit-call connection because of no idle trunk being found in the outgoing link of the switching node Ni when it acts as a transit node, or receiving the transit-call-blocking signal from a transit node when the switching node Ni acts as an originating node; and a completion/blocking signal transfer operation 22*c* of sending the call-completion signal or call-blocking signal to an originating node when the switching node Ni acts as a terminating node, relaying the call-completion signal or call-blocking signal to an originating node when the switching node Ni acts as a transit node, or receiving the call-completion signal or call-blocking signal when the switching node Ni acts as an originating node. The call processing 23 includes: an outgoing trunk selecting operation 23*a* for connecting a call to an idle trunk of a desired link in response to a call-connection request; a trunk holding operation 23*b* for performing a call-connection procedure when receiving the call-completion signal from a terminating node; a call-information transfer operation 23*c* for selecting an appropriate route to the terminating node in response to the call-connection request and a call-blocking operation 23*d* for performing a call-blocking procedure when the call connection to the intended terminating node in response to a call-connection request has finally been blocked. The routing management 24 has databases 24A and functions 24B. The databases 24A include: available alternate routes 24*a*, i.e. the aforementioned sets of available alternate routes received from the network control center NC; cur-

rently assigned alternate routes **24b** selected from the set of available alternate routes **24a;** unavailable alternate routes **24c** selected from the currently assigned alternate routes **24b;** outgoing-trunk-status information **24d** indicating the number of trunks provided in each outgoing link of the switching node Ni; and trunk-status information **24e** indicating the busy/idle status of the trunks of each link. The functions **24B** includes an assigned alternate route initialization/updating function **24f** of determining and updating the assigned alternate routes, a function **24g** of setting the assigned alternate routes available/unavailable and a trunk-status observing function **24h.**

Let it be assumed that the switching nodes, for example, N1 and N4 are an originating and a terminating node in the telecommunications network shown in FIG. 1. In general, the most economical route L14 is selected as the first route, and when no idle trunk is found in the link L14, an alternate route is used. In this instance, possible alternate routes are R134, R164, R124, and R154, but the network control center NC specifies and indicates in advance to the switching node N1 a set of available alternate routes for each first route as shown in Table I of FIG. 3. The available alternate routes to the switching node N4 are routes R134, R154 and R164 which pass through transit nodes N3, N5 and N6, respectively. Based on trunk status information of each outgoing link of the transit nodes N3, N5 and N6 (i.e. the second link of each available alternate route) the switching node N1 selects in advance from the set of available alternate routes at least one route which is expected to be high in the call-completion probability, the alternate route or routes thus selected being assigned as shown in Table I. The switching node N1 selects one of the assigned alternate routes and tries a call connection.

FIGS. 4A, 4B and 4C are flowcharts showing call-connection procedures which each switching node performs, FIG. 4A showing a process flow primarily for an originating node, FIG. 4B a process flow for a terminating node, and FIG. 4C a process flow for a transit node.

In FIG. 4A, upon detection of a call, the switching node identifies the type of the call in step S_1, and if it is a terminating call to the switching node, the process shifts to the process flow shown in FIG. 4B. The switching node checks in step S_{B1} whether or not a trunk to a subscriber or local node is idle which is the destination of the call, and if the trunk is idle, the switching node connects the call to the subscriber (or local node) in step S_{B2} and then sends a call-completion signal back to the originating node in step S_{B3}. Where the trunk to the subscriber or local node (hereinafter referred to as a subscriber trunk, for the sake of brevity) is busy in step S_{B1}, the switching node sends a call-blocking signal back to the originating node in step S_{B4}.

Where it is determined in step S_1 in FIG. 4A that the call is an alternate call, the switching node performs the processing as a transit node, shown in FIG. 4C. In step S_{C1} it is determined whether there is an idle trunk in the outgoing link to the terminating node which is the destination of the call, and if the idle trunk is found, the call is connected to the terminating node through the idle trunk in step S_{C2}. Thus the call is sent to the terminating node, which performs the processing shown in FIG. 4B; namely, the terminating node sends a call-completion or call-blocking signal back to the transit node in step S_{B3} or S_{B4}. The transit node receives the call-completion or call-blocking signal from the terminating node in step S_{C3} and, in step S_{C4}, sends the received signal to the originating node together with trunk-status information of the aforementioned outgoing link of the transit node. Where no idle trunk is found in the outgoing link in step S_{C1}, a call-blocking signal and a transit-call-blocking signal (also referred to as trunk-busy signal) indicating the occurrence of call blocking in the transit node are sent back to the originating node in step S_{C5}. The transit-call-blocking signal is used as trunk status information.

Where it is detected in step S_1 in FIG. 4A that the call is an originating call, the switching node performs the following processing as an originating node. The following description will be given on the assumption that the switching nodes N1 and N4 are an originating and a terminating node, respectively, as in the above. It is checked in step S_2 whether or not there is an idle trunk in the outgoing link L14 which forms the first route to the terminating node, and if an idle trunk is found, the call is connected to the next node via the first route L14 in step S_3. Thus the call is sent to the terminating node N4, which performs the processing shown in FIG. 4B and from which a call-completion or call-blocking signal is sent back to the originating node N1 in step S_{B3} or S_{B4}. The originating node N1 receives the call-completion or call-blocking signal in step S_4 in FIG. 4A, and it is determined in step S_5 which signal was received. Where the received signal is the call-completion signal, the originating node N1 transfers call-information to the terminating node N4 in step S_6 and completes the call-connection procedure. Where it is determined in step S_5 that the received signal is the call-blocking signal, the process terminates with a call-blocking operation in step S_7. When no idle trunk is found in step S_2, the process proceeds to step S_8, wherein an available alternate route, for instance, R134 is selected from the currently assigned alternate routes R134, R154 and R164 for the first route L14, shown in Table I of FIG. 3. Then it is checked whether or not there is an idle trunk in the first alternate link L13 of the selected alternate route R134 in step S_9.

In step S_8, one of the assigned alternate routes is selected randomly, cyclically, or on a predetermined order basis out of currently assigned alternate routes. There are two methods to determine busy/idle trunk status. One method permits the use of the trunk when there is at least one idle trunk. The other one permits the use of the trunk only when there is a predetermined number of two or more idle trunk. The latter method is employed to give the connection of a call using the link as the first route (which call will hereinafter be referred to as a basic call) high priority over the connection of an alternate call.

If an idle trunk can be found in step S_9, the process proceeds to step S_{10}, wherein the call is connected to the next node, e.g. a transit node N3. Thus the call is sent to the transit node N3, wherein the process shown in FIG. 4C is performed. The signal sent back from the transit node N3 in step S_{C4} or S_{C5} is received by the originating node N1 in step S_{11}, and it is checked in step S_{12} whether the signal received in step S_{11} is a call-connection or call-blocking signal. In the case of the call-blocking signal, the call-blocking operation is performed in step S_{13}, and it is checked in step S_{14} whether or not the call-blocking signal is appended with a transit-call-blocking signal, i.e. a trunk-busy signal. The transit-call-blocking signal means that no idle trunk was found in an outgoing link L34 of the transit node N3,

and the assigned route R134 which passes through the transit node N3 is set unavailable in step S_{15}. Then it is checked in step S_{16} whether or not the currently assigned alternate routes need to be updated, and if so, the currently assigned alternate routes are updated in step S_{17}.

The updating of the currently assigned alternate routes in step S_{16} is required in the case (a) where all the currently assigned alternate routes are unavailable, (b) where the number of currently assigned alternate routes set available is smaller than a predetermined value, or (c) where at least one of the currently assigned alternate routes is unavailable. In the case (a), all the currently assigned alternate routes are updated in step S_{17}. In the case (b) or (c), all the currently assigned alternate routes or unavailable ones of them need only to be updated in step S_{17}. Where it is determined in step S_{16} that no updating is needed, the procedure ends.

When it is determined in step S_{12} that the received signal is the call-completion signal, this means that the call has been connected to an idle trunk of the outgoing link L34 in the transit node N3. In this instance, the call-information is transferred to the terminating node N4 via the transit node N3 in step S_{18}, and on the basis of the trunk-status information of the outgoing link L34 in the transit node N3, appended to the received signal, it is checked in step S_{19} whether or not the alternate route R134 needs to be set unavailable. That is to say, in the case where, as a result of the connection of the call to an idle trunk of the outgoing link L34, no more idle trunks exist the number of remaining idle trunks becomes smaller than a predetermined value, or the idle trunk ratio becomes smaller than a predetermined value, the alternate route R134 is set temporarily unavailable in step S_{15}, and then the process proceeds to step S_{16}. Even if it is determined in step S_{19} that the alternate route R134 need not be set temporarily unavailable, it is checked in step S_{16} whether or not the currently assigned alternate routes need to be updated, because there is the possibility that the number of currently assigned alternate routes becomes smaller than a predetermined value.

When it is determined in step S_9 that no idle trunk is found in the first alternate link L13 of the alternate route R134, the currently assigned alternate route R134 is set unavailable temporarily in step S_{20}. Then it is checked in step S_{21} whether or not there still remain any other currently assigned alternate routes which are available, and if yes, the process returns to step S_8, repeating the processing of steps S_8 to S_{21}. When it is determined in step S_{21} that the currently assigned alternate routes are all unavailable, they are all updated in step S_{22} and the procedure ends after the call-blocking operation in step S_{23}. Incidentally, the updating of the currently assigned alternate routes in step S_{22} is performed by the same operation as used in step S_{17}.

When the currently assigned alternate routes are all unavailable in step S_{21}, there is another method. In this method, it is possible to keep the call call-waiting in the broken-linked step S_{24}, all the currently assigned alternate routes are updated in step S_{22} and then it is determine in the broken-line step S_{25} whether to retry the connection of the call held call-waiting. If it is determined to retry the call-connection, the process goes back to step S_8 as indicated by the broken line, trying the call-connection to one of the updated currently assigned alternate routes. If it is determined in step S_{25} not to retry the call-connection, the call-blocking operation is carried out in step S_{23}. This improves the call-completion probability. The return of the process from step S_{25} to S_8 for retrying the call-connection is limited to only once, for example.

There are two methods of setting the selected alternate route of the currently assigned ones routes temporarily unavailable in step S_{15} in FIG. 4A. First, the currently assigned alternative routes are set unavailable for a predetermined time period from the time set in step S_{15} in the process flow of the originating node (in FIG. 4A) or for a time period determined according to the trunk-status information received from the transit node. Second, the transit node sends back the trunk-status information to the originating node together with information of its observation time in step S_{C4} in the process flow of the transit node (in FIG. 4C) and the originating node sets the currently assigned alternate routes unavailable for a predetermined time period from the trunk-status observation time or for a time period determined according to the trunk-status information. In either, case, the time at which each alternate route is released from the unavailable status is calculated in step S_{15} and is stored as shown in Table II of FIG. 5. In step S_8 one of the alternate routes which have already been released from the unavailable status at the current time is selected by referring to Table II of FIG. 5.

The aforementioned trunk-status information which determines the unavailable-status period of the currently assigned alternate routes is, for instance, the number of idle trunks, and the smaller the number of idle trunks, the longer the unavailable-status period is set. For example, when the number of idle trunks is zero, the unavailable-status period is set to 15 seconds, and when two or more trunks are idle, the unavailable-status period is zero seconds. Since the trunk status of links is usually ever-changing, the method of setting the unavailable-status period on the basis of the aforementioned trunk-status observation time is advantageous in that the unavailable-status period of the alternate routes can be set independently of a trunk-status information transfer delay between switching nodes, the waiting time from the observation of the trunk status in the transit node to the transmission of status information, and their variations.

In step S_{17} of FIG. 4A, a required number of new assigned alternate routes are chosen from a set of available alternate routes randomly, in a predetermined cyclic order, or on a predetermined order basis, or alternate routes to be removed from the currently assigned status are set unassignable for a predetermined time period in the same manner as setting the currently assigned alternate routes unavailable as described previously with respect to Table II of FIG. 5 and a required number of new assigned alternate routes are chosen from assignable ones of the set of available alternate routes randomly, in a predetermined cyclic order, or on a prefixed-priority basis.

In the process flow of the originating node described previously in connection with FIG. 4A, one or more available alternate routes selected from the set of available alternate routes specified by the network control center NC are assigned in advance, and in the case of performing alternate routing to comply with a call-connection request, one of the assigned available alternate routes is selected for the call-connection, but it is also possible to perform a call-connection which does not involve such assignment of available alternate routes. An example of such call-connection will be described

below with reference to a process flow shown in FIG. 6.

The process flow in FIG. 6 is a process flow of the originating node and steps shown correspond to steps S_8 through S_{22} in FIG. 4A. Steps S_1 through S_7 in the process flow in FIG. 6 are not shown, because they are identical with steps S_1 through S_7 depicted in FIG. 4A. Furthermore, the process flows of the terminating node and the transit node are the same as the flows shown in FIGS. 4B and 4C, respectively. When no idle trunk is found in the first route in response to a call-connection request, it is checked in step S_8 in FIG. 6 whether or not there are available alternate routes which have idle trunks in their outgoing links, and if not, the process ends with the call-blocking operation in step S_9. When available alternate routes having idle trunks in their outgoing links are found in step S_9, one of such available alternate routes is selected based on the latest trunk-status information (idle-trunk-number information in this example) obtained for each available alternate route, such as shown in Table III in FIG. 7. This is followed by the call-connection operation through the selected available alternate route (i.e. holding an idle trunk and sending the call to the transit node) in step S_{11}. In step S_{12} the originating node receives a call-completion or call-blocking signal and idle-trunk-number information from the transit node. In step S_{13} the idle-trunk-number information of the available alternate route selected in step S_{10}, shown in Table III in FIG. 7, is updated based on the latest idle-trunk information received from the transit node. In step S_{14} it is checked whether the received signal is a call-completion or call-blocking signal. If the signal is the call-blocking signal, the call-blocking operation is performed in step S_{15}, and if the signal is the call-completion signal, the call information is transferred to the next node in step S_{16}. In either case, the process ends. If necessary, step S_{17} is provided between steps S_{14} and S_{15} for checking whether or not to retry the call-connection, as indicated by the broken line, and if the call-connection is to be retried, the process returns to step S_{10}, repeating the above-mentioned processing.

A description will be given of two typical methods for selecting an available alternate route in step S_{10} of FIG. 6.

According to a first method, for example, the transit node sends idle-trunk information, as the trunk-status information, to the originating node in step S_{C4} in the process flow of FIG. 4C. The idle-trunk information may be the busy/idle trunk-status, the number of idle trunks, or the trunk usage; in this example, the number of idle trunks is used as the idle-trunk information. In step S_{12} in the process flow of FIG. 6, the originating node receives from the transit node the idle-trunk information on the selected available alternate route and, in step S_{13}, updates the number of idle trunks corresponding to the available alternate route, shown in Table III of FIG. 7, as described previously. When the process of a call-connection has reached step S_{10}, the originating node refers to Table III and selects an available alternate route having the largest number of idle trunks. When there are two or more available alternate routes of the greatest number of idle trunks, one of them is selected randomly, cyclically, or on a predetermined order basis. Also in the case where binary information indicating the busy/idle trunk-status is used as the above-mentioned idle-trunk information, the available alternate route is selected in the same manner as mentioned above.

According to the second method, the idle-trunk information received in step S_{12} in the first method is used to determine the choice probability (described later) of the available alternate route. The choice probability thus determined is stored as shown in Table III of FIG. 7 and this data is updated according to the received idle-trunk information. When the process of the call-connection has reached step S_{10}, the originating node refers to Table III of FIG. 7 and selects an available alternate route in accordance with the choice probability determined for each available alternate route. Also in this instance, steps S_{14} through S_{17}, S_{19}, S_{20} and S_{22} in FIG. 4A are omitted. One possible method for determining the choice probability is as follows:

Where the idle-trunk information is the number of idle trunks, the choice probability of an available alternate route larger in the number of idle trunks is determined to be higher. Assuming that the numbers of idle trunks of the available alternate routes R134, R154 and R164 are 3, 5 and 2 as shown in Table III of FIG. 7, the choice probabilities of these available alternate routes are determined so that $3/(3+5+2)=0.3$, $5/(3+5+2)=0.5$ and $2/(3+5+2)=0.2$, respectively. With this method, however, when the number of idle trunks of any one of the available alternate routes is zero, its choice probability becomes zero and the available alternate route will never be selected; so that a certain number (0.1, for example) is added to each of the above number of idle trunks.

According to the above-mentioned first method for selecting an available alternate route in step S_{10}, in each switching node an available alternate route of a larger number of idle trunks at each time point is selected. According to the second method, the probability of the available alternate route of a larger number of idle trunks being chosen increases. Consequently, the throughput of the entire network can be improved because the disturbance of the numbers of idle trunks of all the available alternate routes is decreased.

In steps S_2 through S_{23} of the process flow of the originating node shown in FIG. 4A and in their various modifications mentioned above, all alternate routes of the set of available alternate routes are also possible to be assigned. This is substantially equivalent to selecting alternate routes directly from the set of available alternate routes without employing the assignment system. In this case, steps S_{16}, S_{17} and S_{22} in FIG. 4A are unnecessary.

It is also possible to employ a method in which the network control center NC handles single-link routes (first routes) and two-link routes (alternate routes) as equally selectable routes without making a distinction between them and determines sets of available routes for each switching node instead of determining sets of available alternate routes. In this instance, the sets of available routes do not always include single-link routes. In the processes shown in steps S_8 through S_{23} in FIG. 4A and their aforementioned modified examples the originating node selects a route from the set of available routes by the same processing as described previously and tries a call-connection. FIG. 8 shows an example of the process flow of the call-connection procedure by the originating node. The process flow in FIG. 8 is identical with that in FIG. 4A except that steps S_2 through S_7 are left out. In step S_2 the originating node responds to a call-connection request to select that one

of assigned available routes which is not in the unavailable status, thereafter performing the same call-connection procedure as in the case of FIG. 4A. No description will be given of the procedure, for the sake of brevity.

In any of the above-described various route selection algorithms of the present invention for the call-connection procedure of each switching node, assigned available routes in the set of available routes are updated in accordance with their trunk status, by which is increased the probability of selecting an available route which has a large number of idle trunks relative to the other available routes at the time point of occurrence of a call-connection request, and consequently, the call-completion probability is also improved. Moreover, trunk resources of the entire network are used efficiently, and consequently, the network throughput of the entire network increases.

While in the above the transit node has been described to send the trunk-status information to the originating node together with the call-completion or call-blocking signal, the trunk-status information may be sent as a signal independent of the call-completion or call-blocking signal.

Network Control Center

FIG. 9 is a function-block-chart of the network control center NC. The network control center NC performs network-control-data transfer operations **25** for transferring a set of available routes or set of available alternate routes to each switching node at a preset time and routing data generating operations **26** for determining, on the basis of collected data, a set of available routes which are recommended for connecting each switching node-pair and has network databases **27** for preparing the set of available routes.

The network-control-data transfer operations **25** include: a data transmission scheduling operation **25a** for scheduling the transmission of a prepared set of available routes to each switching node; a routing data sending operation **25b** for sending the sets of available routes at the scheduled time; and a network data receiving operation **25c** for receiving network data from each switching node. The routing data generating operations **26** include a possible-route-picking-out operation **26a** for picking out all possible routes through which each switching node-pair in the telecommunications network can be connected, and a set-of-available-routes determining operation **26b** for selecting a set of preferable available routes from the picked-out possible routes on the basis of the network data such as traffic data and trunk data. The network databases **27** include: route data **27a** on the possible routes picked out; transmission system data **27b** for managing the transmission system that constitutes the telecommunications network; trunk data **27c** for managing the number of trunks of each link; and traffic data **27d** for estimating and forecasting the traffic volume which will occur between an originating and a terminating node in the telecommunications network.

The traffic data **27d** is used to estimate the traffic volume between originating and terminating nodes in each time zone or slot of a day. The following four methods can be employed for this estimation.

(a) Traffic data obtained in the past is stored and the traffic volume between each originating and terminating node-pair is calculated statistically based on traffic data obtained in the same time zone of observations days having similar attributes. The attributes of the observation day are those which are likely to influence the traffic, such as weekdays, holidays, days preceding and following consecutive holidays, consecutive holidays, seasons, etc., and this estimation is carried out using a multi-variable analysis considering such attributes.

(b) The traffic volume is estimated using a time-series analysis based on periodically observed traffic data.

(c) The traffic volume is estimated by the combined use of the above-mentioned methods (a) and (b).

(d) The traffic volume is estimated and forecast based on the network operator's experience and knowledge.

Based on the traffic volume in each time zone estimated in accordance with the traffic data **27d**, the time at which the set of available routes is to be sent is determined by the data transmission scheduling operation **25a**. This data transmission time is adaptively changed in accordance with weekdays, holidays, seasons, etc. throughout the year.

The trunk data **27c** includes data on the network topology (i.e. the connections between respective switching nodes through links), the number of trunks of each link and first routes between originating and terminating nodes, and similar data on the constitution of the telecommunications network.

For collecting from each switching node the traffic data **27d** for observing the traffic volume and the trunk data **27c** for updating the trunk-status information of each link, there are a method in which the network control center NC collects the data from each switching node, a method in which the data is transferred from a data collecting system (not shown) provided separately of the network control center NC for implementing the present invention, and a method in which the data is transferred from a dedicated system already employed in the telecommunications network of each country. Such a dedicated system already put into practical use is, for example, a traffic data/trunk-status data collecting system (referred to as ATOMICS (Advanced Traffic Observation and Management Information Collecting System)) used in NTT telecommunications network of Japan. Such a dedicated system and the network control center may also be combined into a network control system.

The following description will be given in connection with the case of producing sets of available alternate routes as the sets of available routes to be sent from the network control center NC to each switching node.

FIG. 10 shows, by way of example, the traffic conditions in the telecommunications network depicted in FIG. 1. The network control center NC is not shown in FIG. 10. A value added to each link represents, in terms of a margin of traffic volume and an overflow traffic volume, the total traffic volume between each switching node-pair on the assumption that the traffic volume has been offered only to the first route therebetween. The margin of traffic volume and the overflow traffic volume are defined as follows:

Overflow traffic volume: Traffic volume having overflowed from the first route

More Specifically, the overflow traffic volume O[i,j] of the link Lij is defined by the following equation:

$$O[i,j] = A_0[i,j] \cdot E(A_0[i,j], N[i,j])$$

where $A_0[i,j]$ is the offered traffic volume on the link Lij, N[i,j] is the number of trunks of the link Lij, and E{*,*} is the Erlang's B equation (or referred to as a loss equation).

Margin of traffic volume: Traffic volume which can be offered until a reference call-connection quality is reached in the case where the first route satisfies the reference call-connection quality

The margin of traffic volume C[i,j] of the link Lij is defined by the following equation, for example:

$$C[i,j]=max\{\bar{A}[i,j]-A_0[i,j],\ 0\}$$

where $\bar{A}[i,j]$ is a value which satisfies $E\{\bar{A}[i,j], N[i,j]\}=B_0$, $A_0[i,j]$ is the basic volume on the link Lij, B_0 is a standard of loss probability (usually $B_0=0.01$), and max{a,b} is a function which takes a larger one of a and b.

The margin of traffic volume C[i,j] and the overflow traffic volume O[i,j] calculated by the above definitions both take values greater than zero, and these values can be calculated for any link Lij. In general, where either one of the overflow traffic volume and the margin of traffic volume is sufficiently larger, the other assumes a value close to zero. In FIG. 10 only the larger one of the overflow traffic volume and the margin of traffic volume is shown for each link and the value of the other is regarded as zero and is not shown for the sake of brevity. In FIG. 10 the margin of traffic volume is indicated by a symbol "−" on its numerical value and the overflow traffic volume by a symbol "+" on its numerical value.

Now, consider the first route between the switching nodes N1 and N4, i.e. a link L14, and the first route between the switching nodes N2 and N3, i.e. a link L23 in FIG. 10. The overflow traffic volumes from the links L14 and L23 are 8 and 7, respectively, and it is necessary to search available alternate routes for alternate call-connections. The criterion for selecting such an available alternate route is the margin of traffic volume through two links which form the alternate route, and the traffic volume which can be offered to the alternate route is determined by the smaller one of the margins of traffic volume on the two links.

The set-of-available-route determining operation *26b* is to determine the set of available routes for each preset time zone by calculating the overflow traffic volume and the margin of traffic volume for each first route based on the trunk data *27c* and the traffic data *27d*. There are the following criteria for obtaining sets of available alternate routes for all first routes through a heuristic iterative calculation. (a) The traffic volume that is overflowed from all the alternate routes between originating and terminating node pair will hereinafter be referred to as a blocked traffic load. A set of available alternate routes which minimize the blocked traffic load between the originating and terminating node-pair of which the blocked traffic load is maximum are selected. (b) Sets of available alternate routes which maximizes the throughput of the entire network. (c) A set of available alternate routes are selected which maximize the call-completion probability between the originating and terminating node-pair of which the call-connection probability is the worst of all the pairs.

FIG. 11 shows a process flow for determining sets of available alternate routes through a heuristic calculation based on the above-mentioned criterion (a).

In FIG. 11, the process starts with the input of the traffic data *27d* and the trunk data *27c* in step S_1, and in step S_2 all alternate routes possible for each link used as the first route are picked out based on the transmission system data *27b* and the trunk data *27c*. In step S_3 a basic traffic volume assignment procedure is performed in which the total traffic volume, which is offered between each originating-terminating node pair in the communications network, is entirely assigned to the first route between the originating-terminating node pair. In the next step S_4 the margin of traffic volume and the overflow traffic volume of each link are calculated, followed by selecting a link of the largest overflow traffic volume in step S_5. The link thus selected will hereinafter be referred to as a first route. Of all alternate routes for the selected first route, an alternate route of the largest margin of traffic volume through two links is selected in step S_6. The alternate route thus selected is stored as an available alternate route corresponding to the first route. Next, in step S_7 a unit volume out of the overflow traffic volume from the selected first route is assigned to the available alternate route selected in step S_6. In the next step S_8 the data of the overflow traffic volume and the margin of traffic volume of each link are recalculated. Steps S_5 through S_8 are repeated until a required number of available alternate routes are determined for each link.

In step S_7 the assignment of unit volume from the overflow traffic volume to the margin of traffic volume can be approximated by a simple method in which the overflow traffic volume from the first route is reduced by the unit volume assigned to the available alternate route and the margin of traffic volume of each link constituting the selected alternate route is decreased by unit volume.

With reference to FIGS. 12A through 12F, a concrete example of sequentially determining available alternate routes by repeating steps S_5 through S_8 will be described using a simple network model with five switching nodes. Five circles indicate switching nodes N1 to N5. In FIGS. 12A through 12F reference numerals L12, L13, L14, L15, L23, . . . of links which interconnect the switching nodes N1 to N5 are omitted, and reference numerals R132, R142, . . . of two-link routes are also omitted. Let it be assumed that the following rules are applied to the procedure for sequentially determining available alternate routes in this simple model.

Rule 1: Where two or more links of the largest overflow traffic volume are found in Step S_5, one of the links, except those for which a required number of available alternate routes have already been determined, is selected randomly.

Rule 2: Where in step S_6 a required number of available alternate routes have already been determined for the link selected in step S_5, a route is selected from these available alternate routes.

Rule 3: Where in step S_6 a required number of available alternate routes have not been determined yet for the link selected in step S_5, a route is selected from the available alternate routes already determined.

Rule 4: Where two or more routes of the largest margin of traffic volume are found in step S_6, one of them is selected randomly.

Rule 5: Where the overflow traffic volume is smaller than unit traffic volume in step S_7, the total overflow traffic volume is assigned to the selected alternate route.

Rule 6: Where the overflow traffic volume 15 is zero in step S_7, a traffic volume 0 is assigned to the selected alternate route.

In FIG. 12A the numeral attached to each link represents the margin of traffic volume or overflow traffic volume calculated in steps S_1 through S_4 of FIG. 11. In the following processing the number of available alternate routes set for each link is 2 for the links L12 and L34 and 0 for the other links, and the unit traffic volume of assignment is 3.

In FIG. 12A, since the link L34 connecting the switching nodes N3 and N4 has the largest overflow traffic volume 9, the link L34 is selected in step S_5 of FIG. 11, and since the alternate route for the link L34 which has the largest margin of traffic volume 7 is R324, the route R324 is determined as an available alternate route of the link L34 in step S_6. In step S_7 a unit volume of 3 out of the overflow traffic volume 9 of the link L34 is assigned to the margins of traffic volume 6 and 7 of the links L23 and L24 which form the route R324. Since the assignment in step S_7 is conducted by addition/subtraction in this example, the overflow traffic volume of the link L34 becomes 6 and the margin of traffic volume of the links L23 and L24 becomes 3 and 4, respectively, and the results of the reassignment are such as shown in FIG. 12B.

Then the process returns to step S_5, wherein the link L34 is selected, which still has the largest overflow traffic volume 6 in FIG. 12B. In step S_6 an alternate route which has the largest margin of traffic volume for the link L34 is selected, and in this case, a route R314 is determined as a second available alternate route of the link L34 in accordance with Rule 3. In step S_7 the unit volume 3 of the current overflow traffic volume 6 of the link L34 is assigned to each of the margins 2 and 5 of links L13 and L14 which form the route R134. The results of updating the data in step S_8 are such as shown in FIG. 12C.

The process returns to step S_5, wherein the link L34 of the largest overflow traffic volume 3 in FIG. 12C is selected, and in step S_6 a route which has the largest margin of traffic volume for the link L34 is selected. In this instance, since two available alternate routes have already been determined for the link L34, the route R324 is selected in accordance with the Rule 2. In step S_7 the unit volume 3 of the current overflow traffic volume 3 of the link L34 is assigned to each of the margins 3 and 4 of the links L23 and L24 which form the route R324. The results of updating the data in step S_8 are such as shown in FIG. 12D.

The process returns to step S_5, wherein a link L12 of the largest overflow traffic volume 1 in FIG. 12D is selected, and in step S_6 a route R152 which has the largest margin of traffic volume for the link L12 is determined as an available alternate route of the link L12. In step S7 the overflow traffic volume 1 of the link L12 is assigned, in accordance with Rule 5, to each of the margins of traffic volume 2 and 3 of the links L15 and L25 which form the route R152. The results of updating the data in step S_8 are such as shown in FIG. 12E.

Then the process returns to step S_5, wherein the link L12 is selected following Rule 1, and in step S_6 the route R152 is determined as a second available alternate route of the link L12 in accordance with Rule 4. In step S_7 the overflow traffic volume 0 is assigned to the route R152, following Rule 6. The results of updating the data in step S_8 are shown in FIG. 12F (which happens to be identical with FIG. 12E). Thus the two available alternate routes set for the links L12 and L34 are determined, with which the process ends.

If the number of alternate routes is predetermined for each set of available alternate routes as explained above, there is the possibility that all the links with overflow traffic volume or all the alternate routes with the margin of traffic volume are gone before the predetermined number of available alternate routes are determined. In the former case, the unit volume for assignment is reduced so that the overflow traffic volume can be assigned to all available alternate routes. In the latter case, when no alternate route with the margin of traffic volume is found in step S_6 in the process flow described above, an alternate route which is the smallest in the overflow traffic volume through two links is selected.

Furthermore, the alternate route which is used for the actual call-connection is selected by the state-dependent adaptive routing which is executed by each switching node; and consequently, if the number of available alternate routes in the set of available alternate routes is selected larger than usual, unpredictable conditions such as a trunk failure and a traffic variation can be dealt with sufficiently.

FIG. 13 shows a process flow for determining a set of available alternate routes by a heuristic iterative calculation which will maximize the entire throughput of the network, referred to previously in item (b). Steps S_1 through S_4 in this process flow are identical with those shown in FIG. 11, and in these steps the overflow traffic volume and the margin of traffic volume of each link are calculated.

Based on the following rules a possible alternate route of the largest margin of traffic volume is selected in step S_5.

Rule 1: Where a required number of available alternate routes have already been obtained for the first route concerning the alternate route, and the overflow traffic volume of the first route is zero or the alternate route concerned is not included in the set of available alternate routes already obtained, the alternate route is not selected.

Rule 2: Where the required number of available alternate routes have not been obtained yet for the first route concerning the alternate route and the overflow traffic volume of the first route is zero and the alternate route concerned is included in the available alternate routes, the alternate route is not selected.

In step S_6. The possible alternate route selected in the preceding step S_5 is stored as an available alternate route for the first route.

In step S_7. The unit volume of the overflow traffic volume from the first route is assigned to the selected possible alternate route.

In step S_8: The overflow traffic volume of the first route and the margin of traffic volume of each link on the selected possible alternate route are updated.

The above-mentioned steps S_5 through S_8 are repeated until the required number of available alternate routes are determined for each link. According to the process flow shown in FIG. 13, since the overflow traffic in the entire network is assigned efficiently so that the margin traffic in the entire network is used up as much as possible, the sets of available alternate routes are determined which maximize the throughput of the network.

FIG. 14 shows a process flow for determining the sets of available alternate routes by a heuristic iterative calculation, using as the criterion the call-completion probability mentioned previously in item (c).

In steps S_1 through S_3 traffic volume is assigned to the first routes between each originating and terminating node-pair in the network on the basis of the traffic data and the trunk data of all links as in the case of FIG. 11. In step S_4 the call-completion probability of a basic call is calculated for each link, and the overflow traffic volume and the margin of traffic volume of each link are calculated as in step S_4 in FIG. 11. The call-completion probability γ of the link Lij is expressed by $\gamma=1-B[i,j]$ and the call-blocking probability B[i,j] of the link Lij can be obtained by the following simultaneous equations:

$$A[i,j] = A_0[i,j] + \sum_{j \in R[i,j]} \frac{A_0[i,k] \cdot B[i,k]}{|R[i,j]|} + \sum_{j \in R[k,j]} \frac{A_0[k,j] \cdot B[k,j]}{|R[k,j]|}$$

$$B[i,j] = E(A[i,j], N[i,j])$$

where A[i,j] and A_0[i,j] are the offered traffic volume and the basic traffic volume of the link Lij, R[i,j] and |R[i,j]| are the set of available alternate routes and the number of available alternate routes for the link Lij, k is the number representing a transit node Nk, E is the Erlang's B equation, and N[i,j] is the number of trunks of the link Lij.

In the next step S_5 a link of the lowest call-completion probability is selected, and in step S_6 one of possible alternate routes which has the highest call-completion probability when the selected link is used as the first route is selected as an available alternate route. It is assumed, however, that the call-completion probability of the alternate route is given by the lower one of the call-completion probabilities of the two links which form the alternate route. In step S_7 the unit volume of the overflow traffic of the selected link of the lowest call-completion probability is assigned to two links of the above-mentioned alternate route of the highest call-connection probability. In step S_8 the traffic volumes which are applied two links of the alternate route are updated based on the assigned traffic volume, and the call-completion probability of the basic call on each of the links is calculated based on the updated traffic volume. Steps S_5 through S_8 are repeated until a required number of available alternate routes are selected for each link.

As will be appreciated from the first route selecting procedure in step S_5 in FIG. 14, the criterion for obtaining an appropriate sets of available alternate routes in this process flow is to determine a set of available alternate routes which minimizes the blocked traffic load between an originating-terminating node pair which is the largest in the traffic volume which cannot be carried by all of the afore-mentioned available alternate routes. In order for all users to utilize the telecommunications network at the same grade of service, it may be desirable to employ a set of available alternate routes which minimizes the call-completion probability between an originating and terminating node-pair which is the lowest in terms of the call-connection quality therebetween as mentioned previously in connection with the process flow shown in FIG. 14.

While in the above a predetermined number of available alternate routes are determined for each link through the heuristic iterative calculation as described previously in respect of FIGS. 11, 13 and 14, it is also possible to determine the set of available alternate routes by continuing the heuristic iterative calculation until a certain condition has been satisfied, instead of predetermining the number of available alternate routes for each link. A description will be given, with reference to FIG. 15, of process flow in which the heuristic iterative calculation is performed for determining the sets of available alternate routes, using the throughput of the network as a criterion.

In the process flow shown in FIG. 15 steps S_1 through S_4 are identical with those in FIGS. 11 and 13, and in these steps the overflow traffic volume and the margin of traffic volume are calculated for each link.

In step S_5 a possible alternate route of the largest margin of traffic volume is selected as in the case of FIG. 13, but this selection is made following the rule mentioned below.

Rule 1: Where the overflow traffic volume of the first route corresponding to the possible alternate route is zero, the alternate route is not selected. The alternate route selected in step S_5 is stored as an available alternate route for the above-mentioned first route in step S_6 as in the case of FIG. 13. In the next step S_7 the unit volume of the overflow traffic volume from the first route is assigned to the selected alternate route, and in step S_8 the overflow traffic volume of the first route and the margin of traffic volume of each link on the selected alternate route are recalculated and updated. In the next step S_9 the throughput of the entire network is calculated and its value is stored, at the same time, corresponding to the sets of available alternate routes having already been determined.

Steps S_5 through S_9 are repeated until the overflow traffic volume of every link is reduced down to zero, and for each repetition of these steps one available alternate route for any one of the links is added and the throughput of the network corresponding to the sets of available alternate routes at that time point is obtained.

When the overflow traffic volumes of all the links are reduced to zero, the process proceeds to step S_{10}, in which the largest one of the throughput values, each obtained upon each repetition of steps S_5 through S_9, is found and the set of available alternate routes determined at the time point at which the largest throughput was obtained is finally determined as the intended set of available alternate routes.

The calculation of the throughput of the entire network in step S_9 is conducted by the following method, for instance. Letting T[i,j] represent the carried traffic volume from an originating node Ni to a terminating node Nj, the throughput P is given by the following equation:

$$P = \sum_{i,j \in V} T[i,j]$$

and the carried traffic volume T[i,j] is given by the following equation:

$$T[i,j] = (1 - B[i,j]) \cdot A_0[i,j] + \sum_{k \in R[i,j]} \left(\frac{A_0[i,j] \cdot B[i,j]}{|R[i,j]|} \times (1 - B[i,j])(1 - B[k,j]) \right)$$

Here, V is the set of switching node in the network, B[i,j] is the call-blocking probability of the link Lij, A_0[i,j] is the basic traffic volume on the link Lij, R[i,j] is

the set of available alternate routes of the link Lij, |R[i,j]| is the number of available alternate routes, and k is the number indicating a transit node Nk.

FIGS. 16 and 17 are simulation results respectively showing the call-completion probability against the number of available alternate routes and the adaptability to actual traffic variations using the model in which the set of available alternate routes were picked out following the process flow shown in FIG. 11.

In FIGS. 16 and 17 there are shown the results of computer-simulation in the case where the state-dependent dynamic routing was performed using the set of available alternate routes picked out following the processing procedure of the network control center. The network model used for the evaluation is a mesh network with 36 switching nodes, in which a designed traffic volume between each originating-terminating node-pair is 30 erl and the offered traffic volume therebetween is 30 erl on the average; hence the network is set in an unbalanced traffic condition in which the traffic volume is randomly set based on the unit distribution. Consequently, first routes of large overflow traffic and first routes of large margin of traffic volume are distributed with each other in the network.

The vertical axis in FIG. 16 represents the worst call-completion probability between the origin-destination node pair, and the horizontal axis represents the number of available alternate routes provided equally for each first route. In FIG. 16 the characteristic (a) is obtained in the case where a limited number of available alternate routes were provided for each switching node in accordance with the procedure of the present invention and the characteristic (b) is obtained in the case where all alternate routes were applied to each switching node, that is, in the case of the conventional state-dependent dynamic routing by centralized control in each switching node. From the results, it is found that the call-completion probability in the case of limiting the number of available alternate routes is higher. Moreover, since the characteristic (a) varies gently with an increase in the number of available alternate routes, the number of available alternate routes can be determined within the range in which the maximum level of (a) is maintained. As a result of this, it is possible to enhance adaptability to unpredictable conditions such as a traffic prediction error and a trunk failure.

FIG. 17 shows the capability of maintaining performance in an unpredictable condition such as the above-mentioned traffic prediction error or trunk failure, that is, against a prediction error. The vertical axis represents the worst call-completion probability between the, origin-destination node pair, and the horizontal axis represents a traffic prediction error ratio between an origin-destination node pair which is used for picking out the sets of available alternate routes, that is, an error ratio between the actual traffic volume and the predicted one. When the number of available alternate routes is too small or too large, the level of the call-completion probability lowers, yet, when the number of available alternate routes is too large, the capability of maintaining performance against the prediction error improves, because it is difficult to be affected by the traffic prediction error. In other words, it is seen that when the number of available alternate routes is 8, the call-connection probability is sufficiently high and the capability of maintaining performance under equipment failure is also sufficiently high as shown in FIG. 17.

In the FIGS. 11, 13, 14 and 15 the process of determining the set of available alternate routes in the network control center NC a plurality of routes between each originating-terminating node-pair are divided into the process of determining the first route which has high priority and the process of determining alternate routes which are used when the traffic volume of the first route overflows, but the present invention can be applied, of course, to the method to determine a set of available routes from the routes between the origin-destination nodes, without dividing them into the first route and alternate routes. As for the process flow in this case, assuming that a virtual first route having no idle trunk is provided between each node-pair separately of all possible routes including a single-link route, the procedure for assigning traffic to the first route in step S_3 in the process flow of FIG. 11 can be implemented by assigning to the virtual first routes the traffic offered between the node pair. The procedure for selecting the link of the largest overflow traffic volume in step S_5 can also be implemented by selecting that one of the virtual first routes which has the largest overflow traffic volume.

Although in the above the routing control method of the present invention has been described as being applied to a telecommunications network, the routing control method of the present invention can be applied as well to a telecommunications network in which links connected via a communications satellite (hereinafter referred to as communications satellite links) can be selected as alternate routes. An example of such a telecommunications network will be described with reference to FIG. 18.

In FIG. 18 five switching nodes N1 through N5 are interconnected via links L12, L13, L14, L23, . . . (which are referred to also as ground links), and each switching node can be connected to the other switching nodes via a communications satellite CS by communications satellite links S12, S13, . . . indicated by the broken lines. For the sake of clarity, no network control center is shown. In the communications network containing the links for interconnecting the switching nodes via the communications satellite, a communications satellite link is used as an alternate route for trying a call-connection only in the case where each cannot perform the call-connection via the first route and no idle trunk is found in any outgoing ground links of currently assigned available alternate routes. Assume, for example, that the switching nodes N1 and N2 are an originating and a terminating nodes, respectively, and routes R132 and R142 are the currently assigned available alternate routes. Where no idle trunk is found in the first route L12 and no idle trunk is found in either of the alternate routes R132 and R142, the process passes through, for example, steps S_8, S_9, S_{20} and S_{21} of FIG. 4A twice and through steps S_{24} and S_{22} and then reaches step S_{25} indicated by the broken line. If it is determined in step S_{25} to retry the call-connection, the process does not return to step S_2 but instead it is checked whether a trunk is idle in the communications satellite link S_{12}, and if so, the call is connected to the communications satellite link S12, after which the same processing as in steps S_4 through S_7 are performed. If no idle trunk is found in the communications satellite link S12, the process ends with the call-blocking operation.

Incidentally, transmission systems are not always different with the first route which directly connects two switching nodes in telecommunications networks

are not always formed by a transmission system independent from other links. For example, links L12, L13, L14, L23, L24 and L34 which connect four switching nodes N1, N2, N3 and N4 in FIG. 19 each form the first route, but the link L13 is accommodated in the same hardware transmission systems T12 and T23 as the links L12 and L23. In this instance, however, the link L13 only passes through the switching node N2 and the switching node N2 does not perform the call-connection. When a failure occurs in the transmission system T12 or T23 in such a transmission network, no call-connection is possible even if a certain route is selected from the links L12 and L23 as an alternate route for the link L13 which is the first route. In this case, by including in the set of available alternate routes in advance, as additional alternate routes, links L14 and L34 accommodated in transmission systems T14 and T34 different from those systems T12 and T23 in which the link L13 is accommodated, it is possible to avoid a serious problem of making both of the first route and its alternate routes unavailable, even if a failure occurs in the transmission system T12 or T23. To this end, the network control center may include such significant alternate routes in the set of available, alternate routes in advance, or each switching node may include such significant alternate routes in the set of available alternate routes received from the network control center. The different transmission systems herein mentioned include transmission systems different in a wide sense, such as systems installed using physically different cables passing through different places, a ground transmission system and a communications satellite system, a digital transmission system and an analog transmission system, or a wire transmission system and a radio transmission system.

Although in the above each first route is defined by one link which connects two switching nodes, it may also be defined by a predetermined number of links which connect the two switching nodes. In such an instance, one or more transit nodes are contained in the first route, and two-link alternate routes are defined for each link which constitutes the first route. Also in such a telecommunications network the process flow by each switching node may be substantially the same as the process flow shown in FIG. 4A, for example, and the process flows in the other embodiments may also be used.

As will be appreciated from the description given so far, the present invention has such advantages as follows:

(i) The sets of available alternate routes are sent from the network control center to each switching node, but since the alternate route to be used according to the real-time network status is selected under distributed control of the switching node, the frequency of control between the network control center and the switching node can be reduced markedly as compared with the frequency needed in the state-dependent adaptive routing placed under centralized control of the network control center. The traffic in Japan, for instance, reaches its peak in substantially the same time zone all over the country and two or three times a day. Accordingly, the set of available alternate routes sent from the network control center needs only to be prepared in accordance with the traffic volume in the peak time, and the traffic volume decreases in other time zones as a whole, and hence can be dealt with within the range of the sets of available alternate routes provided in the peak time zone. As a result of this, the sets of available alternate routes needs only to be sent from the network center to each switching node two or three times a day. Furthermore, even if the network control center does not function because of a failure, the switching node searches for the second-best route through use of the set of available alternate routes provided so far, thereby implementing a highly reliable system.

(ii) According to the present invention, since each switching node performs the state-dependent adaptive routing, idle trunks of links in the network which result from traffic variations or mismatching of trunk resources can be utilized more efficiently than in the case of the time-dependent adaptive routing system.

(iii) According to the present invention, since the range of search for routes, i.e. the set of available alternate routes, is limited taking into account the traffic assignment throughout the network, the number of routing failures by each switching node until finding an appropriate route is smaller than in the case of the conventional state-dependent adaptive routing by each switching node. This affords reduction of the amount of the processing by the switching node, and in the case of employing a method in which a call is handled as a blocked call when call-connection is failed in alternate routes, its completion probability can be improved.

(iv) In the state-dependent adaptive routing by each switching node, the node usually manages data on the set of available alternate routes for each first route. In the present invention, however, since the number of available alternate routes is limited, the amount of data to be managed is smaller than in the case of managing the data on alternate routes throughout the network. Moreover, it is necessary to observe the network conditions, from the point of a network operation, such a condition as the transit-call-completion probability in an alternate route for each link on the first route. Also in this case, the present invention reduces the number of counters for measurement and the amount of measured data to be processed, because the number of available alternate routes is limited.

As described above, according to the present invention, the network control center limits the route-search range, taking into account the traffic conditions and the trunk status, and the sets of available alternate routes are sent to each switching node, and each node performs the state-dependent adaptive routing within the range of the sets of available alternate routes. This permit effective use of the idle network resources which result from traffic variations and mismatching of network resources. Moreover, the frequency of control between the network control center and each switching node can be reduced as compared with the frequency of control in the state-dependent adaptive routing under centralized control of the network control center. The number of routing failures until finding an appropriate route by each switching node is smaller than in the case of the state-dependent adaptive routing by the switching node. Besides, the amount of data to be managed in each switching node, the number of counters and the amount of measured data to be processed in the switching node are smaller than in the case of managing data on all routes in the telecommunications network.

It will be apparent that many modifications and variations may be effected without departing from the scope of the novel concepts of the present invention.

What is claimed is:

1. An adaptive routing control method for a telecommunications network in which a plurality of switching nodes are interconnected via links each composed of a plurality of trunks, one or more routes formed by one or more of said links are provided between each node pair made up of two arbitrary ones of said switching nodes, and at least one network control center is connected via a control signal link to each of said switching nodes, said method comprising:

a step wherein said network control center adaptively determines, for each said node pair, a set of available routes composed of routes which are set available based on the traffic volume in said telecommunications network and the trunk status of said links;

a step wherein said network control center sends said set of available routes to each switching node of each said node pair;

a step wherein each said switching node receives and stores said set of available routes sent from said network control center;

a step wherein each said switching node responds to each call-connection request to select one of said routes from said set of available routes and perform a call-connection procedure based on trunk-status information obtained with respect to the most recent call-connections through respective said routes;

a step wherein, when a call requesting said switching node for connection is a call to a transited from one of the other switching nodes which is the originating node of the call to another of said switching nodes which is the terminating node of said call, said switching node acts as a transit node, is connected to said terminating node, and transfers to said originating node trunk-status information of said link which constitutes said selected available route; and

a step wherein upon each reception of said trunk-status information corresponding to said selected available route, said originating node stores and updates said trunk-status information.

2. The method of claim 1 further comprising a step wherein said network control center updates said sets of available routes at a predetermined time and sends said updated sets of available routes to each said switching node.

3. The method of claim 1 wherein each said set of available routes is a set of available alternate routes, composed of one or more alternate routes for a first route which is a predefined one of said routes between each said node pair, and further comprising a step wherein each said switching node responds to each said call-connection request to try to find an idle trunk in said first route preferentially, and a step wherein when having failed in finding an idle trunk in said first route, each said switching node tries to find an idle trunk in one of said alternate routes in said set of available alternate routes.

4. The method of claim 1, 2, or 3 wherein said set of available routes is determined in a manner to satisfy at least one of the following three conditions:

(a) letting a traffic volume overflowing from each said set of available routes be identified as a blocked traffic load, said blocked traffic load between one of said switching node pairs which is larger than said blocked traffic load between any other switching node pairs is minimized approximately;

(b) the throughput throughout said telecommunications network is maximized approximately; and

(c) a call-completion probability between one of said switch-node pairs which is lower than a call-completion probability between any other node pairs is maximized approximately.

5. The method of claim 3 wherein said network control center includes in said set of available alternate routes for each said first route at least one of alternate routes accommodated in a transmission system different from that in which said first route is accommodated.

6. The method of claim 3 wherein each said switching node adds to said set of available alternate routes at least one of alternate routes accommodated in a transmission system different from that in which said first route is accommodated.

7. The method of claim 1 wherein said step of selecting one of said routes from said set of available routes and performing a call-connection by each said switching node includes a step of preselecting one or more available routes from each said set of available routes and assigning said preselected available routes, and a step of responding to a request for the connection of a call to select said one route from said assigned available routes and perform said call-connection procedure.

8. The method of claim 7, further including a step wherein as a result of said call-connection procedure using said selected one of said assigned available routes, at least one more available route is selected from said set of available routes and assigned if one of the following three conditions is satisfied: (a) said call could not be connected, (b) said call could be connected but all trunks in said selected one route have become busy, and (c) said call could be connected but the number of idle trunks remaining in said selected one route has become smaller than a predetermined value.

9. The method of claim 1 wherein said step of selecting one of said routes from said set of available routes and performing a call-connection procedure by each said switching node includes a step of preselecting one or more available routes from said set of available routes and assigning said preselected available routes, and a step of responding to said request for the connection of a call to select a currently available one of said assigned available routes.

10. The method of claim 9 further comprising a step wherein when said trunk-status information received by said originating node indicates a high possibility of a call being blocked in said link connected to said terminating node, said originating node sets said assigned available routes including said link unavailable for a predetermined period of time.

11. The method of claim 10 further comprising a step wherein when the number of those of said assigned available routes which are not unavailable becomes smaller than a predetermined value, said switching node cancels the assignment of at least said assigned available routes having been set unavailable and newly assigns those of said available routes which are assignable.

12. The method of claim 11 further comprising a step of inhibiting the assignment of said assignment-canceled available routes for a predetermined period of time.

13. The method of claim 9 further comprising a step wherein when said trunk-status information received by said originating node indicates a high possibility of a call being blocked in said link connected to said terminating node, said originating node cancels the assignment of said assigned available routes including said link and

inhibits their reassignment for a predetermined period of time, and a step wherein said originating node assigns one of said available routes which are assignable, in place of said assignment-canceled available routes.

14. The method of claim 12 or 13 wherein said predetermined period of time for which the assignment of said assignment-canceled available routes is inhibited is a fixed period of time.

15. The method of claim 12 or 13 wherein said predetermined period of time for which the assignment of said assignment-canceled available routes is inhibited is determined on the basis of said trunk-status information.

16. The method of claim 9 further comprising a step wherein when no idle trunk is found in an outgoing link constituting said assigned available route selected by each said switching node in response to said call connection request, said selected assigned available route is set unavailable for a predetermined period of time.

17. The method of claim 16 further comprising a step wherein when no idle trunk is found in said outgoing link constituting said selected assigned available route, each said switching node repeats said call-connection procedure, using one of the other assigned available routes which are not in an unavailable status.

18. The method of claim 16 or 17 further comprising a step wherein when all of said assigned available routes are unavailable, said switching node cancels their assignments and newly assigns those of said available routes which are assignable,

19. The method of claim 17 further including a step of inhibiting assignment of said assignment-canceled available routes for a predetermined period of time.

20. The method of claim 10 wherein said predetermined period of time for which said assigned available routes are set unavailable is based on the time at which said originating node receives said trunk-status information.

21. The method of claim 10 wherein said switching node for transiting said call transfers the time of observation of the trunk status of said link to said originating node together with said trunk-status information, and based on said received observation time, said originating node sets said assigned available routes unavailable for said predetermined period of time.

22. The method of claim 20 or 21 wherein said predetermined period of time for which said assigned available routes are set unavailable is determined in accordance with said trunk-status information.

23. The method of claim 20 or 23 wherein said predetermined period of time for which said assigned available routes are set unavailable is a fixed period.

24. The method of claim 9 further comprising a step wherein said switching node for transiting said call performs a procedure for connecting said call to said trunk of said link which constitutes said selected assigned available route and is connected to said terminating node, receives from said terminating node a response signal indicating the completion or blocking of said call and sends said response signal to said originating node.

25. The method of claim 24 wherein said switching node for transiting said call appends said trunk-status information to said response signal and sends them to said originating node.

26. The method of claim 24 wherein said switching node for transiting said call sends said trunk-status information to said originating node separately of said response signal.

27. The method of claim 1 wherein said step of selecting one of said available routes includes a step of determining the choice probability of each of said available routes based on the trunk-status information thereof, and a step of selecting one of said available routes based on said choice probability.

28. The method of claim 1 wherein said trunk-status information is the number of idle trunks of each of said links and that one of said available routes which is selected has the largest number of idle trunks.

* * * * *